



TUGAS AKHIR - KI141502

APLIKASI DETEKSI KEMIRIPAN NADA LAGU

Ericko Ruly Hartanto
NRP 5111100 116

Dosen Pembimbing
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

APPLICATION OF MELODY RESEMBLANCE DETECTION

Ericko Ruly Hartanto
NRP 5111100 116

Advisor

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Aplikasi Deteksi Kemiripan Nada Lagu

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

ERICKO RULY HARTANTO

NRP : 5111 100 116

Disetujui oleh Dosen Pembimbing tugas akhir :

Prof. Ir. HANDAYANI TJANDRASA,
M.Sc., Ph.D.
NIP: 194908231976032001

Dr.Eng. CHASTINE FATICHAH,
S.Kom., M.Kom.
NIP: 197512202001122002



(pembimbing 1)

(pembimbing 2)

**SURABAYA
JUNI 2015**

Aplikasi Deteksi Kemiripan Nada Lagu

Nama Mahasiswa : Ericko Ruly Hartanto
NRP : 5111100116
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.
Dosen Pembimbing 2 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.

ABSTRAK

Di era globalisasi ini, seiring dengan berkembangnya kemajuan teknologi, dunia musik di dunia juga berkembang dengan pesat. Hal tersebut dibantu dengan banyaknya aplikasi-aplikasi yang dibuat untuk meningkatkan kualitas dari musik-musik di zaman sekarang serta didorong dengan distribusi musik yang makin mudah dengan adanya fasilitas-fasilitas disediakan di internet. Hal ini bisa berdampak baik pada kemajuan musik di dunia, tetapi bisa juga berdampak buruk. Salah satu dampak buruk yang ada pada saat ini adalah makin maraknya kegiatan plagiarisme nada lagu. Hal ini tentu saja, menimbulkan masalah pada pelanggaran hak cipta. Oleh karena itu dibutuhkan suatu program yang dapat mengidentifikasi plagiarisme nada lagu.

Untuk mengatasi masalah tersebut, maka dibuat suatu program yang dapat mengidentifikasi plagiarisme lagu melalui nada lagu. Akan digunakan suatu metode perhitungan kemiripan yang dimana dalam Information Retrieval, perhitungan kemiripan didefinisikan sebagai metode pembelajaran data untuk menghitung nilai kemiripan dari sekelompok fitur/atribut dari lagu. Tujuan dari Perhitungan kemiripan adalah untuk menemukan seberapa dekat kemiripan antara beberapa atribut/fitur antara fitur-fitur inputan dengan fitur-fitur yang ada pada database.

Dalam tugas akhir ini, digunakan metode Dynamic Time Warping yang memungkinkan untuk mengetahui seberapa dekatkah kemiripan lagu inputan dengan lagu yang ada pada database, sehingga dimungkinkan untuk melakukan pemeringkatan lagu. Untuk melakukan

pengambilan nada lagu pada sebuah lagu maka digunakan suatu metode ekstraksi fitur yaitu YAAPT (Yet Another Algorithm for Pitch Tracking) sehingga atribut-atribut yang dikerjakan hanya berupa nada yang penting saja dimana nada yang terekstraksi akan membentuk suatu pola titinada (pitch) yang menjadi acuan untuk pengecekan kemiripan sebuah lagu. Pendeteksian melalui nada lagu ini dilakukan dengan tujuan untuk mengatasi masalah plagiarism multi-bahasa.

Aplikasi telah dibuat dan kemudian diuji fungsinya. Pengujian dilakukan melalui skenario yang mencerminkan plagiarisme terhadap nada lagu . Hasil ujicoba menunjukkan bahwa tidak semua plagiarisme nada lagu dapat dideteksi dengan baik dengan nilai akurasi terbaik yaitu 83.33%. Aplikasi yang dibuat membutuhkan waktu rata-rata 23.428 menit untuk mendapatkan hasil akhir .

Kata Kunci: Plagiarisme, Pitch, Information Retrieval, Dynamic Time Warping, Yet Another Algorithm for Pitch Tracking.

Application of Melody Resemblance Detection

Student Name : Ericko Ruly Hartanto
Student ID : 5111100116
Major : Teknik Informatika FTIf-ITS
Advisor 1 : Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.
Advisor 2 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.

ABSTRACT

In this globalisation era, both technology and music are rapidly developing. It is due to the fact that there are various applications made specifically to improve the quality of music as well as the online facilities that simplify the distribution of the music itself. While these could bring positive effect to the music industry, they could, unfortunately, also cause negative impacts. One obvious instance is the plagiarism of the melody, tone and pitch of existing songs. In order to prevent the violation of copyright law, a new similarity inspector program is extremely required.

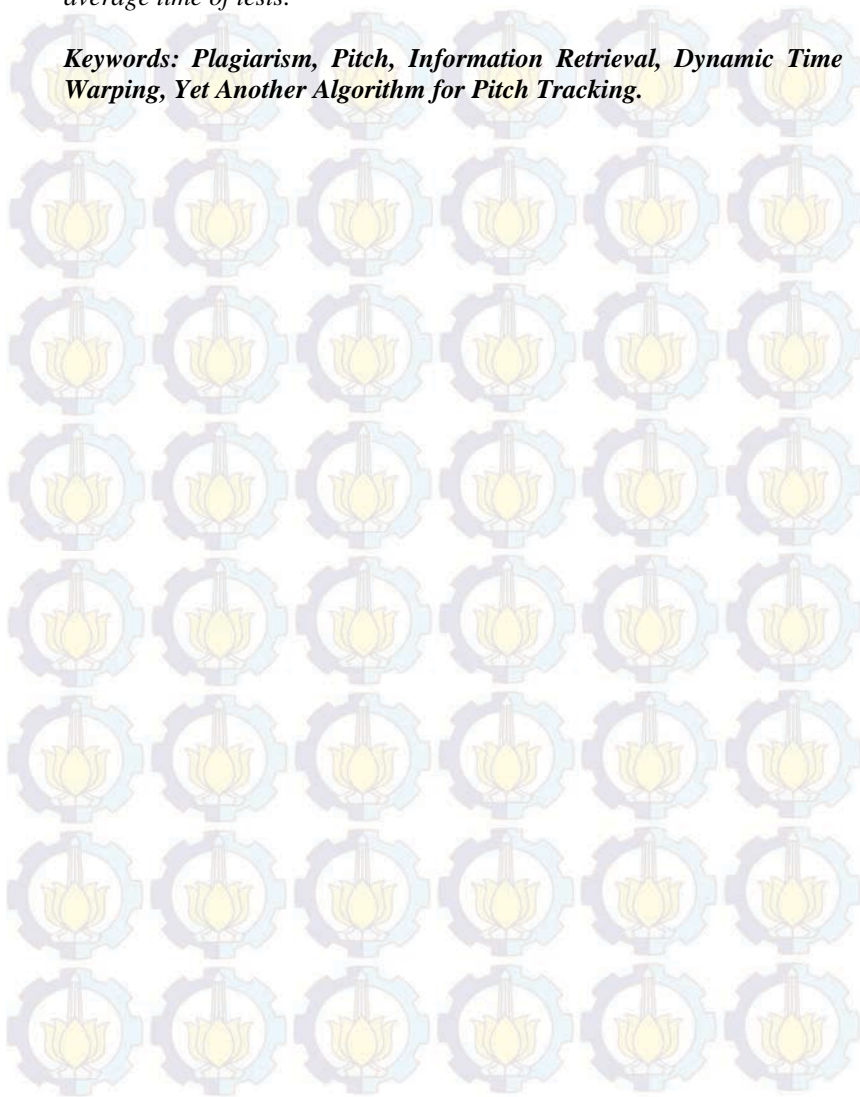
Such application has been designed and engineered to aid this concerning trend. A similarity measure technique method – which according to Information Retrieval means a data-analysing function that quantifies the resemblance between the features or the attributes of two different songs – has been utilised effectively to investigate the resemblance of a certain song to the ones stored in the database.

In this final assignment, song ranking is achieved by utilising Dynamic Time Warping to identify the similarity between the input song and the songs stored in the database. To retrieve a song's melody, a feature-extraction method known as YAAPT (Yet Another Algorithm for Pitch Tracking). These methods will only attribute key melodies, which then a pitch will be extracted and used as the benchmark to check the similarity between songs. This melody detection is required to prevent multi-language plagiarism issues.

The application and its functionalities have successfully passed various musical plagiarism scenarios and tests. The tests indicates that not all musical plagiarism can be properly detected with 83.33% in best

accuracy of tests. Retrieving the final result requires 23.428 minutes in average time of tests.

Keywords: Plagiarism, Pitch, Information Retrieval, Dynamic Time Warping, Yet Another Algorithm for Pitch Tracking.



KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul :

“Aplikasi Deteksi Kemiripan Nada Lagu”

Harapan dari penulis semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Rudiyanto, Ibu Lelyati Djaja Soepitno yang selalu memberikan dukungan penuh dan doa untuk menyelesaikan tugas akhir ini.
2. Kakak Carissa Ruly Komalasari, Adik Michael Ruly Hartanto yang selalu ada untuk memberikan semangat, dan hiburan ketika merasa jenuh.
3. Emak Endang Soedjinawati yang selalu memberikan doa, semangat serta saran dalam mengerjakan tugas akhir ini.
4. Bapak Yusianto, Ibu Ellyati Djaja Soepitno, dan keluarga yang memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
5. Ibu Handayani Tjandra dan Ibu Chastine Fatichah selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
6. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
7. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.

8. Sahabat saya F. Asisi Arifin, Kevin Willyanto, Edwin Autis, dan Bisatya yang selalu ada untuk membantu, memberikan semangat, dan hiburan ketika penulis merasa jenuh.
9. Teman-teman seluruh anggota B-1B dan beberapa C-1B (Ivan Hendrajaya, Steven Fredian A.P., Ebenhaezer W.Y., Andy William, dan Stieven Wirakasa) yang telah memberikan banyak dukungan, saran yang baik dan buruk, serta memberikan semangat kepada penulis.
10. Teman-teman angkatan 2011 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 3,5 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.
11. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2015

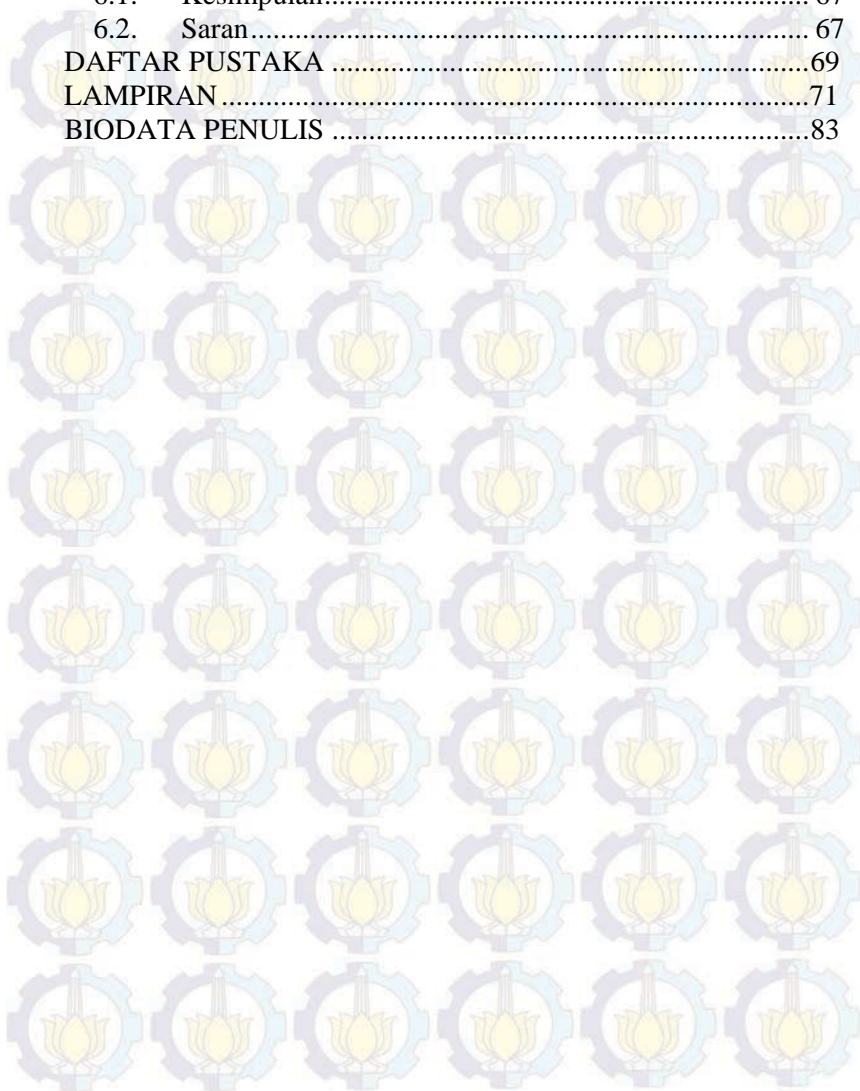
Ericko Ruly Hartanto

DAFTAR ISI

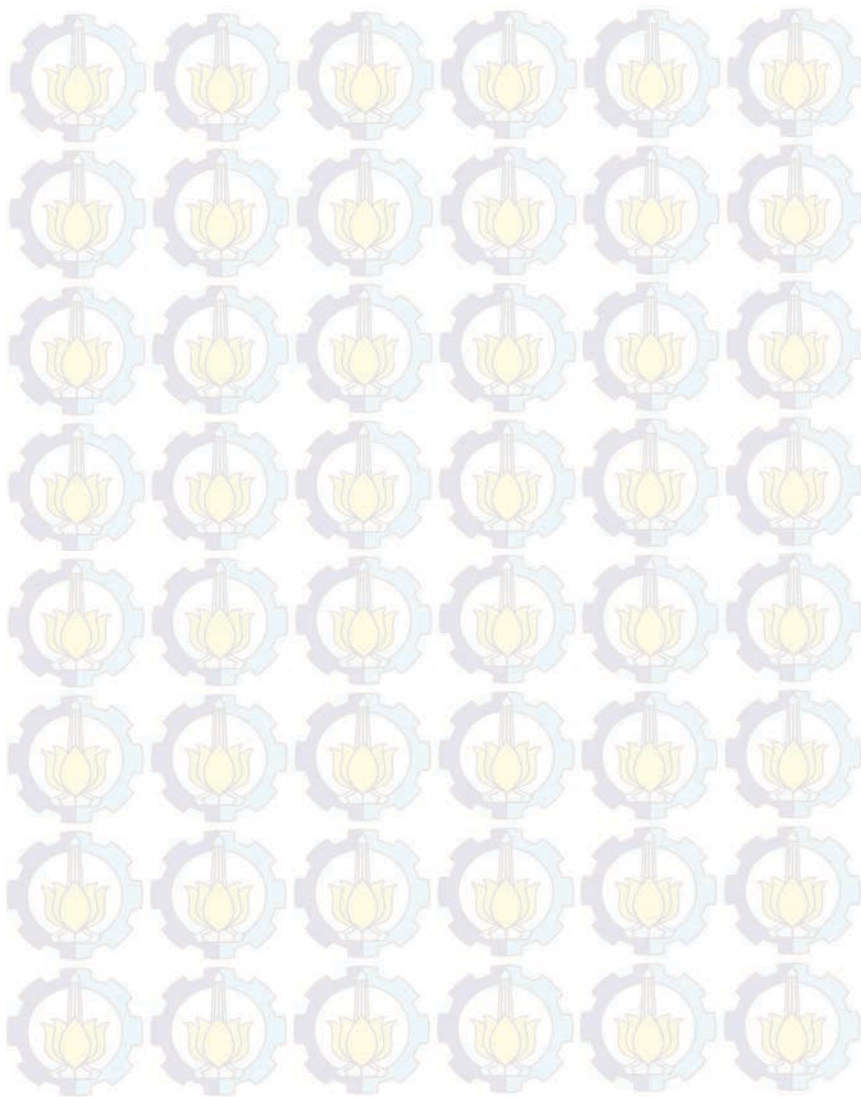
LEMBAR PENGESAHAN	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	2
1.3. Rumusan Permasalahan.....	2
1.4. Batasan Permasalahan	2
1.5. Metodologi	3
1.6. Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1. Sinyal Digital.....	7
2.2. <i>Digital Signal Processing</i>	7
2.3. Melodi	8
2.4. <i>Fast Fourier Transform</i>	9
2.5. <i>Yet Another Algorithm for Pitch Tracking</i>	10
2.5.1. <i>Preprocessing</i>	11
2.5.2. Estimasi Kandidat <i>F0</i> dari Perhitungan <i>Normalize Cross Correlation</i>	12
2.5.3. Perbaikan Pemilihan Kandidat Berdasarkan Informasi Spektral	13
2.5.4. Algoritma <i>Tracking</i>	15
2.6. <i>Dynamic Time Warping</i>	18
BAB III ANALISIS DAN PERANCANGAN SISTEM	21
3.1. Analisis.....	21
3.1.1. Analisis Permasalahan	21
3.1.2. Deskripsi Umum Sistem	22

3.1.3.	Kebutuhan Fungsional	23
3.1.4.	Aktor	23
3.1.5.	Kasus Penggunaan	23
3.2.	Perancangan Sistem.....	28
3.2.1.	Lingkungan Perancangan Perangkat Lunak.....	29
3.2.2.	Perancangan Metode Secara Umum.....	30
3.2.3.	Reduksi Suara Vokal Lagu.....	31
3.2.4.	Eksraksi Fitur Nada Lagu	31
3.2.5.	Re-Dimensi Atribut	32
3.2.6.	<i>Dynamic Time Warping</i>	33
3.2.7.	Normalisasi Hasil <i>Dynamic Time Warping</i>	35
3.2.8.	Perancangan Antarmuka Pengguna	36
BAB IV IMPLEMENTASI.....		39
4.1.	Lingkungan Implementasi	39
4.2.	Implementasi	39
4.2.1.	Implementasi Fungsi wavRead().....	39
4.2.2.	Implementasi Fungsi Reduksi Suara Vokal Lagu.....	40
4.2.3.	Implementasi Fungsi YAAPT()	41
4.2.4.	Implementasi Fungsi fft()	41
4.2.5.	Implementasi Fungsi dtw().....	42
4.2.6.	Implementasi Fungsi Normalisasi Hasil <i>Dynamic Time Warping</i>	42
4.2.7.	Implementasi Fungsi <i>Ranking</i> Nada Lagu	44
BAB V PENGUJIAN DAN EVALUASI		45
5.1.	Lingkungan Pengujian.....	45
5.2.	Dataset dan Data Testing.....	45
5.3.	Skenario Pengujian.....	48
5.3.1.	Pengujian Fungsionalitas <i>Dynamic Time Warping</i>	50
5.3.2.	Pengujian Penghitungan Kemiripan Dengan Inputan Lagu Cover Dari Dataset.....	52
5.3.3.	Pengujian Penghitungan Kemiripan Dengan Inputan Lagu Yang Diopinikan Melakukan Plagiarisasi Terhadap Salah Satu Lagu Pada Dataset	55
5.4.	Analisis dan Evaluasi Hasil Uji Coba	62

BAB VI KESIMPULAN DAN SARAN.....	67
6.1. Kesimpulan.....	67
6.2. Saran.....	67
DAFTAR PUSTAKA	69
LAMPIRAN	71
BIODATA PENULIS	83



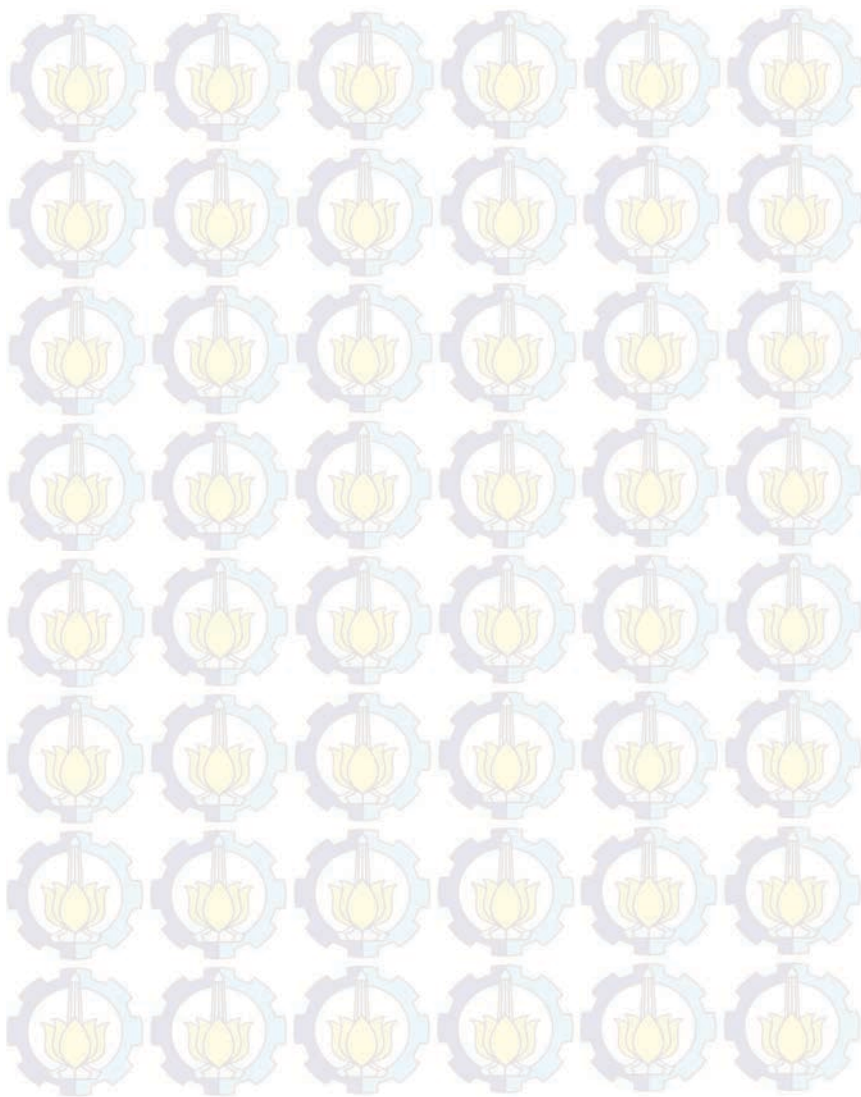
[Halaman ini sengaja dikosongkan]



DAFTAR TABEL

Tabel 4-1. Deskripsi Kebutuhan Fungsional	23
Tabel 4-2. Daftar Kode Diagram Kasus Pengguna	24
Tabel 4-3. Spesifikasi Kasus Penggunaan Memasukkan Lagu Inputan.....	25
Tabel 4-4. Spesifikasi Kasus Penggunaan Melakukan Penghitungan Kemiripan Nada Lagu	27
Tabel 4-5 Lingkungan Perancangan Perangkat Lunak.....	29
Tabel 4-6. Spesifikasi Atribut Antar Muka Tampilan Utama	37
Tabel 5-1. Tabel Data Lagu (1)	46
Tabel 5-2. Tabel Data Lagu (2)	47
Tabel 5-3. Tabel Data Uji Lagu (1).....	47
Tabel 5-4. Tabel Data Uji Lagu (2).....	48
Tabel 5-5. Tabel Hasil Uji Coba Skenario 1	51
Tabel 5-6. Tabel Hasil Uji Coba Skenario 2 (1).....	53
Tabel 5-7. Tabel Hasil Uji Coba Skenario 2 (2).....	54
Tabel 5-8. Tabel Lagu Yang Diopinikan Melakukan Plagiarisme	56
Tabel 5-9. Tabel Hasil Uji Coba Skenario 3 (1).....	57
Tabel 5-10. Tabel Hasil Uji Coba Skenario 3(2).....	58
Tabel 5-11. Tabel Hasil Uji Coba Skenario 3(3).....	59
Tabel 5-12. Tabel Hasil Uji Coba Skenario 3(4).....	60
Tabel 5-13. Tabel Akurasi Uji Coba Dan <i>Running Time</i>	62
Tabel 5-14. Tabel <i>Running Time</i> Tiap Skenario (1).....	62
Tabel 5-15. Tabel <i>Running Time</i> Tiap Skenario (2).....	63
Tabel 5-16. Tabel Akurasi Uji Coba Skenario (1)	64
Tabel 5-17. Tabel Akurasi Uji Coba Skenario (2)	65
Tabel A1-1. Tabel Lagu Yang Diopinikan Melakukan Plagiarisasi	81

[Halaman ini sengaja dikosongkan]



DAFTAR GAMBAR

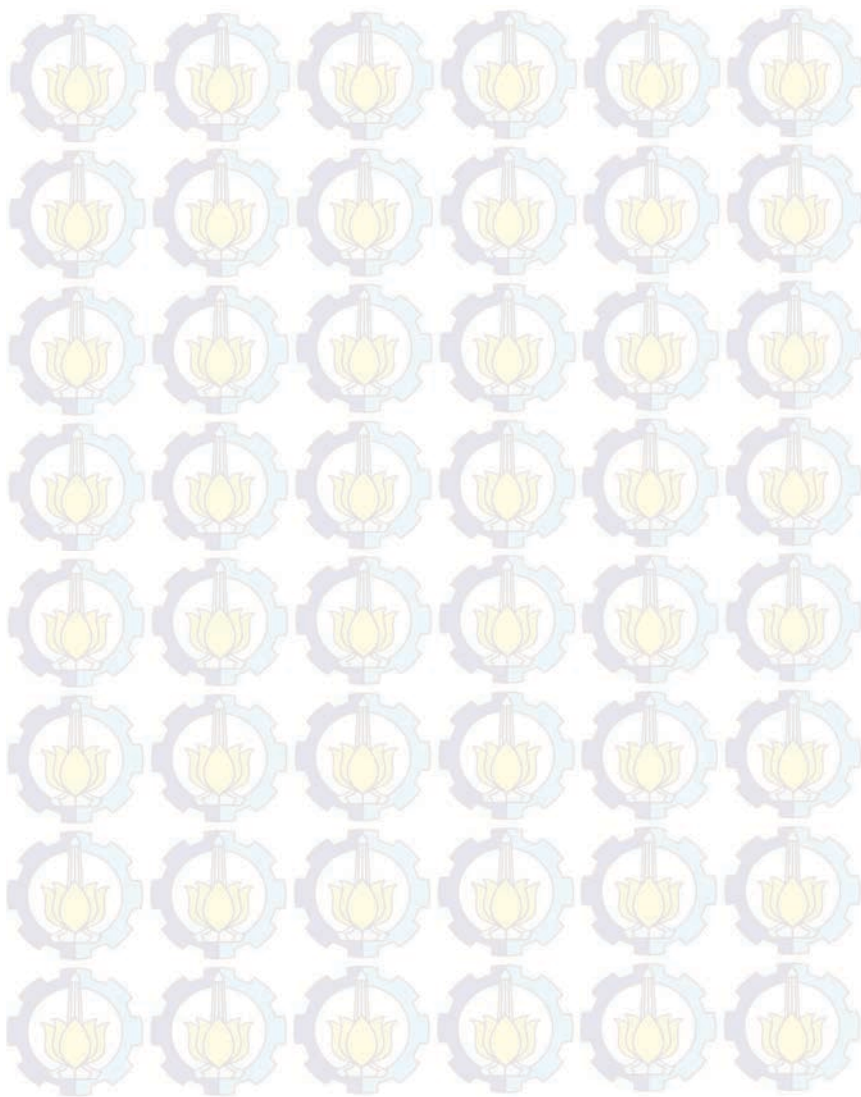
Gambar 2-1. Sinyal Digital	7
Gambar 2-2. Pemrosesan Sinyal Digital	8
Gambar 2-3. Sinyal Suara Akustik Berdasarkan Domain Waktu	11
Gambar 2-4. Spektogram Sinyal Asli.....	11
Gambar 2-5. Spektogram Sinyal Yang Telah Diabsolutkan	12
Gambar 2-6. Sinyal Suara Akustik Berdasarkan Waktu	17
Gambar 2-7. Spektogram Sinyal Akustik.....	17
Gambar 2-8. Kandidat Nada Yang Telah Diperbaiki.....	17
Gambar 2-9. Hasil Dari Pengambilan Nada.....	17
Gambar 2-10. Warping Path.....	19
 Gambar 3-1. Deskripsi Umum Sistem	 22
Gambar 3-2. Diagram Aktivitas Kasus Penggunaan Memasukkan Lagu Inputan	26
Gambar 3-3. Diagram Aktivitas Kasus Penggunaan Melakukan Penghitungan Kemiripan Nada Lagu	28
Gambar 3-4. Diagram Blok Proses Sistem.....	30
Gambar 3-5. <i>Pseudocode</i> Reduksi Suara Vokal Lagu	31
Gambar 3-6. <i>Pseudocode</i> Ekstraksi Fitur Nada Lagu.....	32
Gambar 3-7. <i>Pseudocode</i> Re-dimensi Atribut.....	33
Gambar 3-8. <i>Pseudocode</i> <i>Dynamic Time Warping</i>	34
Gambar 3-9. <i>Pseudocode</i> Normalisasi Hasil <i>Dynamic Time Warping</i>	35
Gambar 3-10. Tampilan Aplikasi Deteksi Kemiripan Nada Lagu.....	36
Gambar 3-11. Tampilan Aplikasi Setelah Input Lagu.....	37
Gambar 3-12. Tampilan Akhir Setelah Proses Perhitungan Kemiripan.....	38
 Gambar A1-1. Hasil Uji Coba Skenario 1 Lagu 1.....	 71
Gambar A1-2. Hasil Uji Coba Skenario 1 Lagu 2.....	71
Gambar A1-3. Hasil Uji Coba Skenario 1 Lagu 3.....	72

Gambar A2-1. Hasil Uji Coba Skenario 2 Lagu 1	72
Gambar A2-2. Hasil Uji Coba Skenario 2 Lagu 2	73
Gambar A2-3. Hasil Uji Coba Skenario 2 Lagu 3	73
Gambar A2-4. Hasil Uji Coba Skenario 2 Lagu 4	74
Gambar A2-5. Hasil Uji Coba Skenario 2 Lagu 5	74
Gambar A3-1. Hasil Uji Coba Skenario 3 Lagu 1.....	75
Gambar A3-2. Hasil Uji Coba Skenario 3 Lagu 2.....	75
Gambar A3-3. Hasil Uji Coba Skenario 3 Lagu 3.....	76
Gambar A3-4. Hasil Uji Coba Skenario 3 Lagu 4.....	76
Gambar A3-5. Hasil Uji Coba Skenario 3 Lagu 5.....	77
Gambar A3-6. Hasil Uji Coba Skenario 3 Lagu 6.....	77
Gambar A3-7. Hasil Uji Coba Skenario 3 Lagu 7.....	78
Gambar A3-8. Hasil Uji Coba Skenario 3 Lagu 8.....	78
Gambar A3-9. Hasil Uji Coba Skenario 3 Lagu 9.....	79
Gambar A3-10. Hasil Uji Coba Skenario 3 Lagu 10.....	79
Gambar A3-11. Hasil Uji Coba Skenario 3 Lagu 11.....	80
Gambar A3-12. Hasil Uji Coba Skenario 3 Lagu 12.....	80

DAFTAR KODE SUMBER

Kode Sumber 4.1. Kode Sumber Fungsi wavRead (1)	40
Kode Sumber 4.2 Kode Sumber Reduksi Suara Vokal Lagu.....	40
Kode Sumber 4.3. Kode Sumber Implementasi Fungsi YAAPT 41	
Kode Sumber 4.4. Kode Sumber Fungsi FFT()	42
Kode Sumber 4.5. Kode Sumber Fungsi dtw (1)	42
Kode Sumber 4.6. Kode Sumber Fungsi dtw (2)	43
Kode Sumber 4.7. Kode Sumber Fungsi Normalisasi.....	43
Kode Sumber 4.8. Kode Sumber Fungsi <i>Ranking</i> Nada Lagu	44

[Halaman ini sengaja dikosongkan]



BIODATA PENULIS



Ericko Ruly Hartanto, lahir di Jombang, pada tanggal 6 November 1993. Penulis menempuh pendidikan mulai dari SDK Santa Maria Immaculata Tabanan, Bali (1999-2005), SMPN 1 Tabanan, Bali (2005-2008), SMAK St.Louis 1 Surabaya (2008-2011) dan S1 Teknik Informatika ITS (2011-2015).

Selama kuliah di teknik informatika ITS, penulis mengambil bidang minat Komputasi Cerdas Visual (KCV). Selama masa perkuliahan penulis pernah mengikuti magang di Perusahaan Gas Negara SBU II Surabaya. Komunikasi dengan penulis dapat melalui email: **erickoruly11@gmail.com**.

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Di era globalisasi ini, seiring dengan berkembangnya kemajuan teknologi, dunia musik di dunia juga berkembang dengan pesat. Hal tersebut dibantu dengan banyaknya aplikasi-aplikasi yang dibuat untuk meningkatkan kualitas dari musik-musik di jaman sekarang serta disokong dengan distribusi musik yang makin mudah dengan adanya fasilitas-fasilitas disediakan di internet. Hal ini bisa berdampak baik pada kemajuan musik di dunia, tetapi bisa juga berdampak buruk. Salah satu dampak buruk yang ada pada saat ini adalah makin maraknya kegiatan plagiarisme nada lagu. Hal ini tentu saja, menimbulkan masalah pada pelanggaran hak cipta. Oleh karena itu dibutuhkan suatu program yang dapat mengidentifikasi plagiarisme nada lagu.

Untuk mengatasi masalah tersebut, maka dibuat suatu program yang dapat mengidentifikasi plagiarisme lagu melalui nada lagu. Akan digunakan suatu metode perhitungan kemiripan dimana dalam *Information Retrieval*, perhitungan kemiripan didefinisikan sebagai metode pembelajaran data untuk menghitung nilai kemiripan dari sekelompok fitur/atribut dari lagu. Tujuan dari Perhitungan kemiripan adalah untuk menemukan seberapa dekat kemiripan antara beberapa atribut/fitur antara fitur-fitur inputan dengan fitur-fitur yang ada pada *database*.

Dalam tugas akhir ini, akan digunakan metode *Dynamic Time Warping* yang memungkinkan untuk mengetahui seberapa dekatkah kemiripan lagu masukan dengan lagu yang ada pada database, sehingga dimungkinkan untuk melakukan pemeringkatan

lagu. Untuk melakukan pengambilan nada lagu pada sebuah lagu maka akan digunakan metode ekstraksi fitur yaitu YAAPT (*Yet Another Algorithm for Pitch Tracking*) sehingga atribut-atribut yang dikerjakan hanya berupa nada yang penting saja dimana nada yang terekstraksi akan membentuk suatu pola titinada (*pitch*) yang menjadi acuan untuk pengecekan kemiripan sebuah lagu. Pendeteksian melalui nada lagu ini dilakukan dengan tujuan untuk mengatasi masalah plagiarism multi-bahasa.

Dengan adanya aplikasi ini, diharapkan untuk pengidentifikasian terhadap plagiarisme semakin mudah dan dapat membantu penegakan hak cipta terhadap sebuah lagu.

1.2. Tujuan

Tujuan tugas akhir ini adalah untuk membangun sebuah aplikasi yang dapat menghitung kemiripan nada sebuah lagu terhadap nada lagu-lagu yang ada dalam *dataset*.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur pada sebuah lagu?
2. Bagaimana melakukan perhitungan kemiripan antara nada lagu inputan dengan nada lagu yang ada pada *dataset*?
3. Bagaimana menguji kinerja aplikasi yang dibangun?

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi ini berbasis *desktop* dengan bahasa pemrograman matlab.
2. Database lagu yang dijadikan data uji beberapa berupa Wav dan apabila belum berupa Wav maka akan di-

convert menjadi Wav dengan perangkat lunak “Audacity”.

3. Lagu yang dimasukkan merupakan lagu yang utuh, yaitu bukan hasil dari pemotongan lagu.
4. *Sampling Frequency* lagu yang digunakan adalah 44100 Hz.
5. Lagu yang digunakan merupakan lagu kualitas bagus yaitu berjenis *stereo*.
6. Tidak memiliki fitur penambahan lagu kedalam dataset serta tidak dapat memutar lagu hasil dari pedeteksian kemiripan.
7. Pengujian skenario plagiarisasi diambil berdasarkan opini yang beredar pada internet.
8. Dataset yang dibuat berasal dari lagu yang telah dilakukan pemrosesan ekstraksi dan re-dimensi dan berjumlah 20 data.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai ekstraksi fitur/ciri dari lagu, serta metode perhitungan kemiripan antara nada lagu yang ada pada *database* dan nada lagu inputan. Mengumpulkan dan menggali informasi dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang dibangun.

2. Analisis dan Perancangan Sistem

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga

didefinisikan pada tahap ini. Pada tahapan ini dibuat desain suatu sistem dan desain proses-proses yang ada

3. Implementasi

Pada tahap ini dilakukan pembuatan elemen perangkat lunak. Sistem yang dibuat berpedoman pada rancangan yang telah dibuat pada proses perancangan dan analisis sistem. Pembangunan aplikasi dilakukan dengan menggunakan kakas bantu *Matlab R2013a* sebagai alat bantu untuk mengimplementasikan kode program. Bahasa pemrograman yang digunakan adalah Matlab.

4. Pengujian dan evaluasi

Pengujian dari aplikasi ini akan dilakukan:

1. Pengujian *Blackbox*

Menurut Rivayi Arifanto(2014) [1], pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak disebut dengan pengujian *blackbox*. Pengujian ini dilakukan untuk menguji apakah lagu-lagu yang diopinikan mirip dengan lagu masukan sudah benar atau belum.

2. Pengujian Tingkat Akurasi

Pengujian tingkat akurasi, digunakan untuk mengukur kemampuan aplikasi dalam melakukan pengecekan kemiripan lagu, seberapa akurat hasil yang dikeluarkan oleh perangkat lunak yang dibuat.

5. Penyusunan buku tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan

- a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
 3. Desain dan Implementasi
 4. Pengujian dan Evaluasi
 5. Kesimpulan dan Saran
 6. Daftar Pustaka

1.6. Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan proses aplikasi, perancangan antar muka sistem, perancangan *dataset* nada lagu.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian *blackbox* dan pengujian tingkat akurasi untuk mengetahui pengujian hasil *similarity calculation*.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

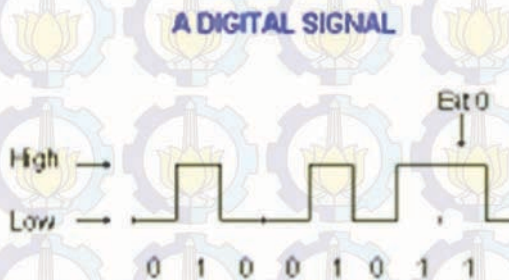
BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut meliputi sinyal digital, *digital signal processing*, melodi, *pitch*, *yet another algorithm for pitch tracking*, *dynamic time warping*.

2.1. Sinyal Digital

Menurut Muhammad Fajar Said(2012) [2], Sinyal data dalam bentuk pulsa yang dapat mengalami perubahan yang tiba-tiba dan mempunyai besaran 0 dan 1, sehingga tidak mudah terpengaruh oleh noise disebut dengan sinyal digital yang pada dasarnya di *code*-kan dalam bentuk biner atau hexa dan besarnya sinyal tersebut dibatasi oleh lebarnya jumlah bit (*bandwidth*) yang mempengaruhi nilai akurasi sinyal tersebut. Gambar sinyal digital digambarkan pada Gambar 2-1.

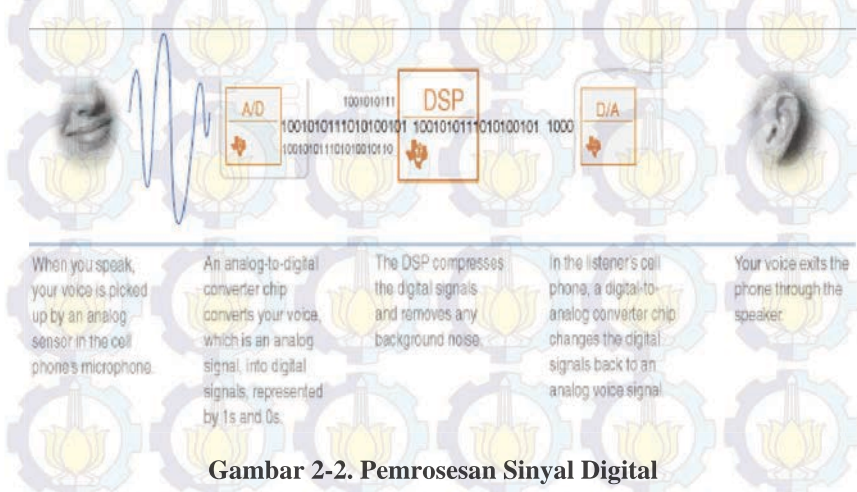


Gambar 2-1. Sinyal Digital

2.2. *Digital Signal Processing*

Menurut Hary Oktavianto (2008) [3], sebuah metode untuk memproses sinyal analog (diwakili oleh rangkaian bilangan) menggunakan teknik matematis untuk melakukan transformasi atau mengambil informasi atau sebuah metode untuk memproses sinyal

analog pada domain digital disebut dengan *Digital Signal Processing*. Contoh dari *Digital Signal Processing* dapat dilihat pada Gambar 2-2.



2.3. Melodi

Menurut Szablcsi (1965) [4], melodi (dari bahasa Yunani $\mu\epsilon\lambda\omicron\delta\iota\alpha$, *melōidia*, “bernyanyi, nyanyian”), dan juga suara, merupakan sebuah suksesi linear dari sebuah nada-nada yang dapat didengar sebagai satu kesatuan. Sebuah melodi juga diartikan sebagai kombinasi dari nada, dan ritme, yang dapat ditambahkan dengan suksesi dari beberapa elemen musik seperti warna suara. Melodi ini biasanya digunakan sebagai latar belakang dari sebuah lagu.

Melodi biasanya terdiri dari satu atau beberapa frase musik atau motif musik, dan biasanya diputar berulang kali dengan komposisi yang berbeda-beda. Melodi juga dapat diartikan sebagai tanda meodi atau nada-nada atau interval dari nada, jarak nada, tarik ulur nada, kontinuitas dan koheren, irama, serta bentuk.

Sehingga dalam pengerjaan tugas akhir ini menggunakan melodi dari lagu yang didefinisikan sebagai sebuah deretan motif

musik yang memiliki titik tinggi rendah (pitch) sehingga dapat digunakan dalam pengecekan kemiripan dari lagu inputan dengan lagu yang ada pada *database*.

2.4. *Fast Fourier Transform*

Menurut Sipasulta (2014) [5] *Fast Fourier Transform* (FFT) yang ditemukan tahun 1965 merupakan pengembangan dari *Fourier Transform* (FT). Penemu FT adalah J. Fourier pada tahun 1822. FT membagi sebuah sinyal menjadi frekuensi yang berbeda-beda dalam fungsi eksponensial yang kompleks.

Definisi FFT adalah metode yang sangat efisien untuk menghitung koefisien dari *Fourier* diskrit ke suatu sekuen *finite* dari data yang kompleks. Karena substansi waktu yang tersimpan lebih dari pada metode konvensional, *fast fourier transform* merupakan aplikasi temuan yang penting didalam sejumlah bidang yang berbeda seperti analisis *spectrum*, *speech* and *optical signal processing*, desain *filter digital*. Algoritma FFT berdasarkan atas prinsip pokok dekomposisi perhitungan *discrete fourier transform* dari suatu sekuen sepanjang N kedalam transformasi diskrit *fourier* secara berturut-turut lebih kecil. Cara prinsip ini diterapkan memimpin ke arah suatu variasi dari algoritma yang berbeda, di mana semuanya memperbandingkan peningkatan kecepatan perhitungan.

Fast Fourier Transform, adalah suatu algoritma untuk menghitung transformasi *fourier* diskrit dengan cepat dan efisien. Karena banyak sinyal-sinyal dalam sistem komunikasi yang bersifat kontinyu, sehingga untuk kasus sinyal kontinyu kita gunakan transformasi *fourier*. Transformasi *Fourier* didefinisikan oleh rumus 2-1.

Dimana $s(f)$ adalah sinyal dalam domain frekuensi (*frequency domain*), $s(t)$ adalah sinyal dalam domain waktu (*time domain*), dan $e^{-j2\pi ft}$ adalah konstanta dari nilai sebuah sinyal, f adalah frekuensi dan t adalah waktu. FFT merupakan salah satu metode untuk transformasi sinyal suara dalam domain waktu menjadi

sinyal dalam domain frekuensi, artinya proses perekaman suara disimpan dalam bentuk digital berupa gelombang *spectrum* suara yang berbasis frekuensi sehingga lebih mudah dalam menganalisa *spectrum* frekuensi suara yang telah direkam.

$$s(f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft} dt \quad (2-1)$$

2.5. Yet Another Algorithm for Pitch Tracking

Yet Another Algorithm for Pitch Tracking (YAAPT) menurut Kavita Kasi (2012) [6] merupakan, sebuah algoritma pendeteksi *pitch* yang handal untuk sinyal digital berupa bunyi, baik suara maupun lagu. *Kernel* dari YAAPT berdasarkan dari algoritma yang diterapkan pada *Robust Algorithm for Pitch Tracking* (RAPT), namun tetap saja kedua algoritma ini berbeda. Perbedaan yang paling menonjol yaitu penggunaan informasi spektografik yang menjadi acuan dalam melakukan pencarian nada. Hanya mengandalkan spektografik saja dapan menimbulkan banyak kesalahan dalam pencarian *F0* pada data-data dengan frekuensi rendah pada spektogram, dimana *F0* didefinisikan sebagai nada dasar dari suatu sinyal suara. YAAPT mengatasinya dengan cara menggabungkan informasi pada spektografik dengan estimasi *pitch* dari metode korelasi, sehingga dapat mendapatkan hasil yang lebih maksimal.

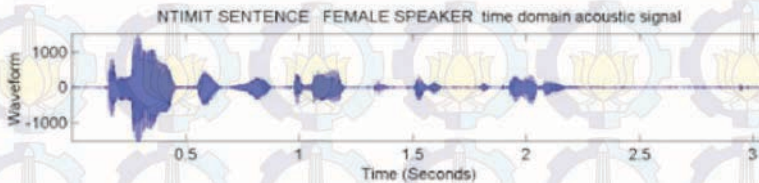
Algoritma ini terdiri dari 5 tahapan utama:

1. *Preprocessing*.
2. Seleksi kandidat *F0* dengan menggunakan *Normalized Cross Correlation* (NCCF).
3. Perbaikan hasil seleksi kandidat berdasarkan informasi spektral.
4. Penentuan *path* akhir dengan menggunakan *dynamic programming*.

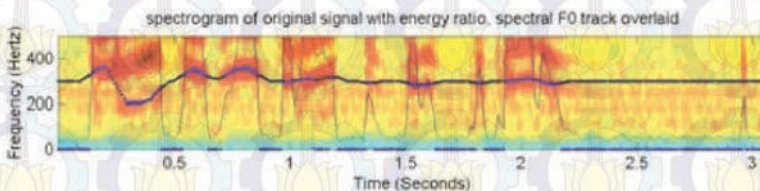
Untuk detailnya akan dijelaskan pada subbab 2.5.1, 2.5.2, 2.5.3, dan 2.5.4.

2.5.1. Preprocessing

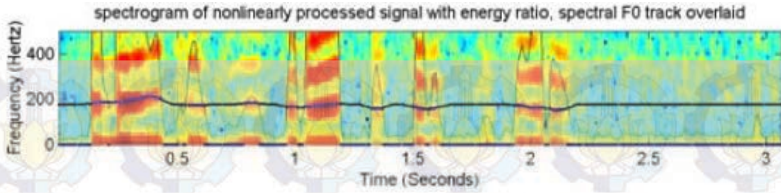
Tahap pertama pada *preprocessing* adalah membuat 2 versi dari sinyal, yaitu sinyal aslinya dan nilai absolut dari sinyal tersebut. Kemudian, tiap sinyal tersebut akan melewati proses *bandpass filter* dengan menggunakan *Hamming Window*. Contoh dari tahap *preprocessing* ditunjukkan pada Gambar 2-3, Gambar 2-4, dan Gambar 2-5. Contoh tersebut diambil dari sinyal percakapan telpon, dimana sinyal akustik ditunjukkan pada Gambar 2-3, *low frequency spectrogram* dari sinyal akustik ditunjukkan pada Gambar 2-4, dan *low frequency spectrogram* dari sinyal yang telah di absolutkan ditunjukkan pada Gambar 2-5. Dua kurva yang melintang pada kedua *spectrogram* akan dijelaskan selanjutnya. Pada contoh gambar *preprocessing* tersebut menunjukkan bahwa F_0 lebih dominan pada bagian bawah *spectrogram* daripada bagian tengahnya. Efek yang sama juga ditemukan pada banyak contoh sinyal, serta beberapa sinyal berkualitas bagus.



Gambar 2-3. Sinyal Suara Akustik Berdasarkan Domain Waktu



Gambar 2-4. Spektogram Sinyal Asli



Gambar2-5. Spektrogram Sinyal Yang Telah Diabsolutkan

2.5.2. Estimasi Kandidat $F0$ dari Perhitungan *Normalize Cross Correlation*

Penggunaan penghitungan nilai korelasi untuk pencarian $F0$ didasarkan pada sinyal yang berkorelasi akan memiliki puncak pada suatu *magnitude* yang besar dalam suatu periode pada suatu nada. Apabila *magnitude* dari nilai puncak bernilai lebih dari 0.6, maka pada *frame* sinyal tersebut biasanya memiliki suara vokal. *Normalized Cross Correlation Function* (NCCF) merupakan modifikasi dari fungsi *autocorrelation* biasa. Rumus untuk penghitungan fungsi korelasi ditunjukkan pada rumus 2-2 dan 2-3.

$$NCCF(k) = \frac{\sum_{n=0}^{N-K} s(n) \times s(n+k)}{e_0 \times e_k} \quad (2-2)$$

$$e_k = \sum_{n=k}^{n=k+N-K} s^2(n), \quad 0 \leq k \leq K-1 \quad (2-3)$$

Dimana $s(n)$ merupakan *frame* sampel suara dan $0 \leq n \leq N-1$.

Menurut David Talkin (1995) [7], *Normalize Cross Correlation Function* cocok digunakan untuk pendeteksian nada dari pada metode *autocorrelation* pada umumnya. Hal ini dikarenakan pada amplitudo sinyal suara, puncak-puncak pada tiap *magnitude*

lebih dominan dan tidak banyak terpengaruh oleh variasi kecepatan dari amplitudo sinyal tersebut.

Tahap awal dari pengambilan puncak untuk penghitungan F_0 yaitu dengan cara memilih titik-titik yang memiliki sisi-sisi bernilai lebih dari 2 dan titik tersebut memiliki nilai lebih dari batas terbawah yaitu 0.3, maka titik tersebut dianggap sebagai puncak. Pada tahap awal ini, dimungkinkan ditemukan banyak nilai puncak. Tahap kedua pada pengambilan puncak yaitu, tiap puncak yang terdeteksi pada tahap awal yang memiliki jeda 2 ms atau lebih akan dieliminasi. Kemudian, setiap nilai puncak yang tersisa akan diberikan nilai signifikansi yang bernilai sebesar *magnitude* dari $NCCF$ pada nilai jedanya. Kemudian pada tahap ketiga, tiap puncak yang memiliki nilai 2 kali dari nilai jedanya, maka nilai signifikansinya akan dinaikkan sebanyak 0.025 dari nilai signifikansi pada tahap kedua, dikarenakan apabila suatu puncak memiliki jeda yang lebih pendek, maka puncak tersebut dianggap sebagai puncak yang benar.

Puncak-puncak dengan nilai signifikansi yang masih tersisa dari ketiga tahap tersebut merupakan kandidat F_0 . Apabila dari 3 tahap diatas tidak ditemukan adanya puncak yang memenuhi syarat, maka sinyal inputan dianggap tidak memiliki suara.

2.5.3. Perbaikan Pemilihan Kandidat Berdasarkan Informasi Spektral

Tiap pola pada spektogram akan memunculkan daerah yang bersuara dan daerah yang tidak memiliki suara, sehingga menunjukkan kontur dari perkiraan F_0 seperti yang dijelaskan oleh Kavita Kasi (2012) [6]. Sinyal asli dan sinyal yang telah diabsolutkan akan ditransformasikan kedalam bentuk spektogram seperti yang telah dijelaskan pada subbab 2.5.1. Pada bagian ini akan dijelaskan metode empiris yang akan digunakan untuk membantu menentukan sinyal yang bersuara atau tidak bersuara, dan metode yang digunakan untuk menentukan kontur nada yang sangat halus/ hampir tidak bersuara dengan kemungkinan *error* yang kecil.

Pemakaian penghitungan *Normalized Low Frequency Energy Ratio* (NFLER) digunakan untuk membantu mengindikasikan daerah sinyal yang bersuara atau tidak bersuara. Jumlah dari nilai-nilai absolut dari sampel spektral (rata-rata energi tiap *frame*) yang memiliki nilai lebih dari daerah frekuensi rendah akan diambil, dan dinormalisasi dengan cara membagi nilai tersebut dengan nilai rata-rata dari energi frekuensi rendah tiap *frame* suara. Rumus dari perhitungan NFLER ditunjukkan pada rumus 2-4.

$$NFLER = \frac{\sum_i x(i,j)}{\frac{1}{N} \sum_i \sum_j x(i,j)} \quad (2-4)$$

- N = Jumlah banyaknya *frame*
 i = Indeks dari frekuensi
 x(i,j) = Log *magnitude* dari frekuensi yang rendah pada spektrogram
 j = Indeks dari *frame*

Secara umum, NFLER akan memiliki nilai yang besar untuk daerah yang memiliki suara, dan bernilai kecil untuk daerah yang tidak memiliki suara, dengan batasan nilai mendekati 0.65 untuk dijadikan acuan untuk bagus tidaknya keputusan dalam penentuan bersuara atau tidaknya suatu daerah/*frame* pada suatu sinyal. Tetapi penentuan bersuara atau tidaknya suatu *frame* pada metode ini tidak akan menggunakan cara umum diatas. Metode yang digunakan seperti pada tahapan berikut :

1. Pembagian frekuensi rendah pada spektrogram di benahi dengan menggunakan *mask* selebar 60 Hz dan sebesar 3 *frames* tiap operasi.
2. Pengambilan puncak secara simpel digunakan untuk menentukan puncak pertama pada daerah cakupan pencarian $F0_{min}$: $F0_{max}$ dengan nilai $F0_{min}$ 60 Hz dan $F0_{max}$ 400 Hz. Apabila tidak ada puncak

ditemukan, maka *frame* tersebut dianggap tidak bersuara.

3. *Track* yang dihasilkan sampai pada tahap kedua dibenahi dengan menggunakan *3-point median filter*.
4. Penghitungan estimasi $F0$ rata-rata dan standar deviasi dari $F0$ dihitung dengan menggunakan nilai sepertiga tengah (*frame* bersuara diurutkan berdasarkan frekuensinya) dari *frame* bersuara dari tahapan ketiga.
5. Semua *frame* yang tidak bersuara diestimasi dari tahapan ketiga, dan semua *frame* tersebut yang memiliki nilai yang jauh berbeda dari nilai rata-rata $f0$, akan digantikan dengan nilai rata-rata 0.
6. *Track* yang dihasilkan pada tahap 5 akan diperbaiki dengan menggunakan *5-point median filter*.
7. Cara heuristik digunakan untuk menggabungkan 2 spektral $F0$ *tracks* untuk menentukan *track* yang telah dibenarkan

Untuk *frame* yang memiliki suara ada sinyal suara, kandidat $F0$ dari NCCF akan dites kedekatannya terhadap titik poin $F0$ pada spektral. Untuk tiap kandidat yang kurang dari $1.5 F0_{min}$ dari nilai $F0$ spektral, maka nilai signifikansinya akan dinaikkan sebanyak 1.25, dan puncak-puncak yang memiliki nilai yang jauh atau lebih dari $1.5 F0_{min}$ akan direduksi atau dihapus.

2.5.4. Algoritma *Tracking*

Hasil akhir dari tahapan estimasi dan perbaikan kandidat $F0$ adalah matriks kandidat $F0$, sebuah matriks signifikansi. Sebuah kurva *NFLER* (dari sinyal asli), dan spektrografik dari *track F0*. Data-data ini akan digunakan untuk ,mendapatkan matriks bobot lokal dan transisi, dimana bobot lokal dari *track* nada untuk semua kandidat yang ada ditentukan dengan menggunakan *dynamic programming*.

Pertama-tama tiap *frame* akan dilakukan *pre-classified* berdasarkan *NFLER* baik yang benar-benar tidak bersuara ($NFLER \leq 0.5$) atau yang kemungkinan bersuara ($NFLER > 0.5$). Jadi, untuk

daerah yang benar-benar tidak bersuara, semua kandidat $F0$ diberi nilai 0. Dan nilai signifikansinya diberi nilai 0.99. Untuk daerah yang kemungkinan bersuara, $F0$ dimasukkan sebagai salah satu kandidat, dimana nilai signifikansinya diberi nilai tengah. Nilai signifikansi dari kandidat yang tidak bersuara akan diberi nilai 1-(nilai signifikansi dari kandidat bersuara yang terbaik).

Setelah semua dari nilai signifikansi telah selesai di hitung, maka bobot lokal akan dihitung dengan rumus:

$local\ cost = 1 - nilai\ signifikansi$	(2- 5)
---	---------------

Dimana perhitungan ini menggunakan operasi satu matriks saja pada Matlab.

Poin utama pada komputasi dari bobot transisi akan dijelaskan pada tahapan-tahapan berikut ini:

1. Untuk tiap pasangan suksesif kandidat bersuara (kandidat $F0$ yang bernilai bukan 0) akan dihitung dengan rumus:

$$transition\ cost(i) = (F0_i \times F0_{i-1})^2 \quad (2- 6)$$

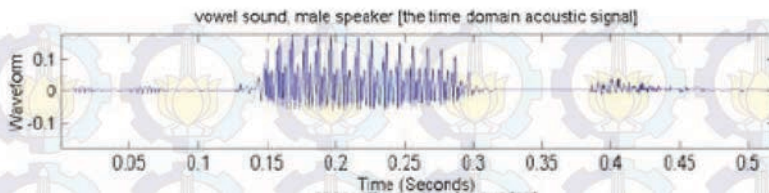
2. Untuk tiap pasang dari kandidat suksesif, yang mana salah satunya bersuara, akan dihitung dengan rumus:

$$transition\ cost(i) = NFLER(unvoiced_frames) \quad (2- 7)$$

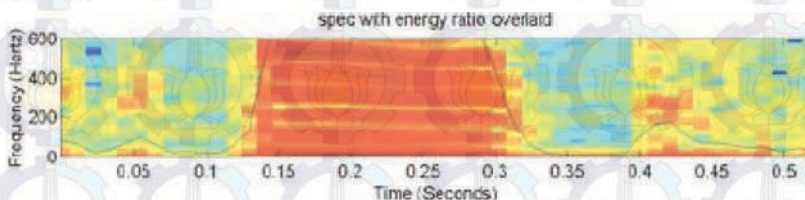
3. Untuk tiap pasang dari kandidat suksesif yang tidak bersuara, akan dihitung dengan rumus:

$$transition\ cost(i) = NFLER_i \times NFLER_{i-1} \quad (2- 8)$$

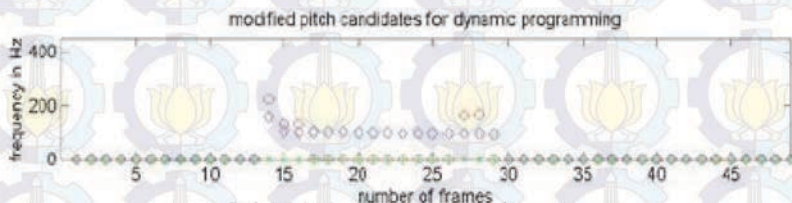
Berbagai macam persamaan telah dijelaskan diatas, ditambah satu konstanta tambahan yang digunakan untuk menyesuaikan rasio dari bobot lokal ke bobot dinamis secara keseluruhan, yang secara empiris ditentukan berdasarkan pengamatan dari beberapa ratus sampel rekaman. Algoritma secara keseluruhan dilustrasikan pada Gambar 2-6, Gambar 2-7, Gambar 2-8, dan Gambar 2-9.



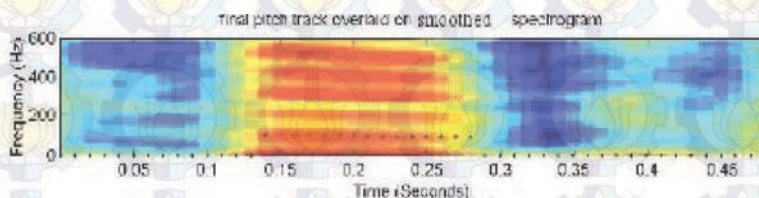
Gambar 2-6. Sinyal Suara Akustik Berdasarkan Waktu



Gambar 2-7. Spektrogram Sinyal Akustik



Gambar 2-8. Kandidat Nada Yang Telah Diperbaiki



Gambar 2-9. Hasil Dari Pengambilan Nada

Semua algoritma yang disebutkan diatas, diimplementasikan dalam fungsi *Matlab*, yang dianggap dapat berintegrasi dengan perangkat lunak lain dengan mudah. Proses utama dalam algoritma

ini adalah perhitungan beberapa kandidat $F0$ dan nilai signifikansi untuk tiap *frame*, proses pengambilan $F0$ dari spektral sinyal, serta proses untuk pengambilan nada secara keseluruhan.

2.6. *Dynamic Time Warping*

Menurut Sakoe , H. & S. Chiba (1978) [8] *Dynamic Time Warping algorithm* (DTW) adalah algoritma yang menghitung *optimal warping path* antara dua waktu. Algoritma ini menghitung baik antara nilai *warping path* dari dua waktu dan jaraknya. Misalnya, kita memiliki dua sekuens numerik (a_1, a_2, \dots, a_m) dan (b_1, b_2, \dots, b_m). Dengan pemisalan ini, maka dapat dikatakan bahwa panjang dua sekuens ini bisa saja berbeda.

Algoritma ini dimulai dengan penghitungan jarak lokal antara elemen dari 2 sekuens menggunakan yang memiliki jarak yang berbeda. Jika dalam matriks maka dapat ditulis dengan memiliki n garis dan m kolom, secara umum:

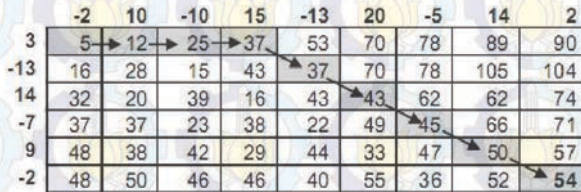
$$d_{ij} = |a_i - b_j|, \quad i = \overline{1, n}, \quad j = \overline{1, m} \quad (2-9)$$

Mulai dengan matrik jarak lokal, kemudian minimum jarak matriks antar sekuens ditentukan menggunakan algoritma program dinamis dan mengikuti kriteria optimasi ditunjukkan pada persamaan (2-10).

$$a_{ij} = d_{ij} + \min (a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1}) \quad (2-10)$$

Dimana a_{ij} merupakan jarak minimal antara subsekuens. *Warping path* adalah sebuah *path* yang melewati jarak matrik minimum dari elemen a_{11} ke a_{nm} . Ongkos *warping path* secara global dari dua sekuens ditunjukkan pada persamaan 2-10.

$$GC = \frac{1}{p} \sum_{i=1}^p w_i \quad (2-11)$$



	-2	10	-10	15	-13	20	-5	14	2
3	5	12	25	37	53	70	78	89	90
-13	16	28	15	43	37	70	78	105	104
14	32	20	39	16	43	43	62	62	74
-7	37	37	23	38	22	49	45	66	71
9	48	38	42	29	44	33	47	50	57
-2	48	50	46	46	40	55	36	52	54

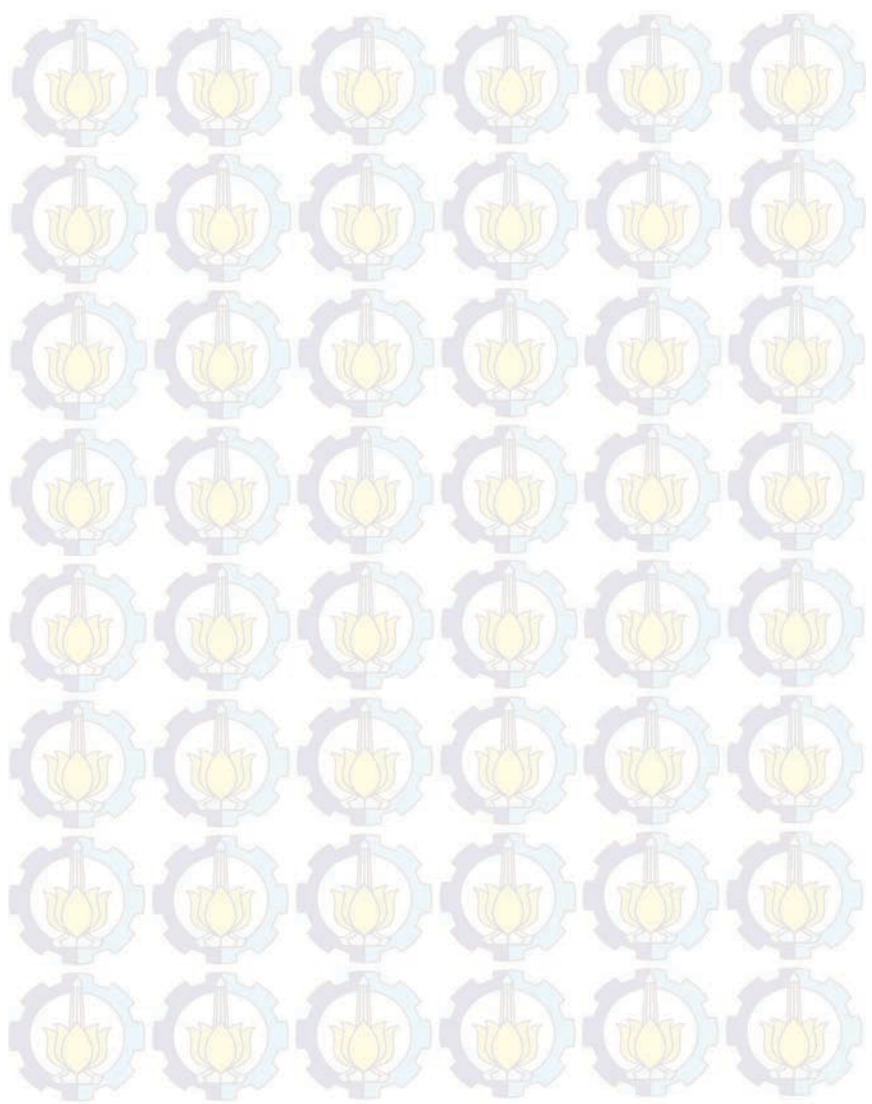
Gambar 2-10. Warping Path

Dimana W_i adalah elemen yang dimiliki *warping path* dan p adalah jumlahnya. Penghitungannya dibuat untuk dua sekuens yang diperlihatkan pada gambar dibawah dan *warping path* diberi *highlight*.

Karena prinsip optimasi dalam program dinamis diimplementasikan pada teknik *backward*, mengidentifikasi *warp path* menggunakan tipe struktur dinamis yang disebut *stack*. Seperti algoritma pemrograman dinamis lainnya. DTW memiliki kompleksitas polinomial. Ketika sekuens memiliki banyak elemen, minimal ada dua ketidaknyamanan:

1. Lamanya waktu pengerjaan mengingat matriks yang besar
2. Memiliki banyak perhitungan jarak

[Halaman ini sengaja dikosongkan]



BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.1.1. Analisis Permasalahan

Makin maraknya kegiatan plagiarisme nada lagu menimbulkan masalah pada pelanggaran hak cipta. Banyaknya artis-artis pembuat lagu protes, karena karyanya dipakai oleh orang lain. Oleh karena itu dibutuhkan suatu aplikasi yang dapat mengidentifikasi plagiarisme nada lagu.

Untuk mengatasi masalah tersebut, maka dibuat suatu program yang dapat mengidentifikasi plagiarisme lagu melalui nada lagu. Pendeteksian melalui nada lagu ini dilakukan dengan tujuan untuk mengatasi masalah plagiarisme multi-bahasa. Aplikasi yang dibangun menggunakan suatu metode perhitungan kemiripan dimana dalam *Information Retrieval*, perhitungan kemiripan didefinisikan sebagai metode pembelajaran data untuk menghitung nilai kemiripan dari sekelompok fitur/atribut dari lagu. Tujuan dari Perhitungan kemiripan adalah untuk menemukan seberapa dekat kemiripan antara beberapa

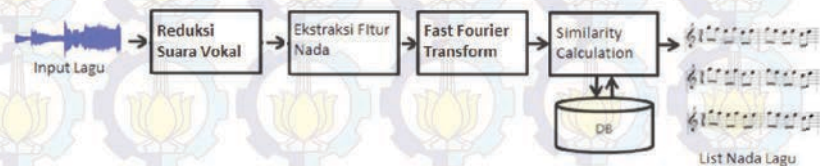
atribut/fitur antara fitur-fitur inputan dengan fitur-fitur yang ada pada *database*.

Dengan adanya program ini, diharapkan untuk pengidentifikasian terhadap plagiarisme semakin mudah dan dapat membantu penegakan hak cipta terhadap sebuah lagu.

3.1.2. Deskripsi Umum Sistem

Aplikasi yang dibangun pada tugas akhir ini merupakan aplikasi pendeteksi kemiripan nada lagu yang menggunakan algoritma *Dynamic Time Warping* untuk penghitungan nilai kemiripan nada lagu, serta menggunakan algoritma *Yet Another Algorithm for Pitch Tracking* untuk pengambilan fitur melodi pada lagu.

Gambar 3-1 merupakan deskripsi umum aplikasi pendeteksi kemiripan nada lagu yang akan dikembangkan. Sesuai gambar, pengguna diberi pilihan untuk memasukkan lagu, apabila lagu telah diinputkan, maka akan muncul tombol *calculate* kemudian apabila pengguna meng-klik tombol *calculate* maka program akan melakukan melodi dengan menggunakan algoritma *Yet Another Algorithm for Pitch Tracking*. Kemudian apabila proses ekstraksi ini sudah selesai, maka sistem akan memulai melakukan re-dimensi terhadap hasil ekstraksi, karena dimensinya terlalu besar yang akan mempengaruhi kecepatan program, kemudian akan dilanjutkan dengan menghitung kemiripan nada lagu inputan dengan nada lagu-lagu yang ada pada dataset yang telah disediakan. Apabila proses ini sudah selesai maka program akan mengeluarkan 5 judul lagu yang memiliki kemiripan paling dekat dengan nada lagu inputan.



Gambar 3-1. Deskripsi Umum Sistem

3.1.3. Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang harus dimiliki sistem. Kebutuhan fungsional mendefinisikan layanan yang harus disediakan dan reaksi sistem terhadap masukan atau pada situasi tertentu. Daftar kebutuhan fungsional dapat dilihat pada Tabel 4-1.

Tabel 4-1. Deskripsi Kebutuhan Fungsional

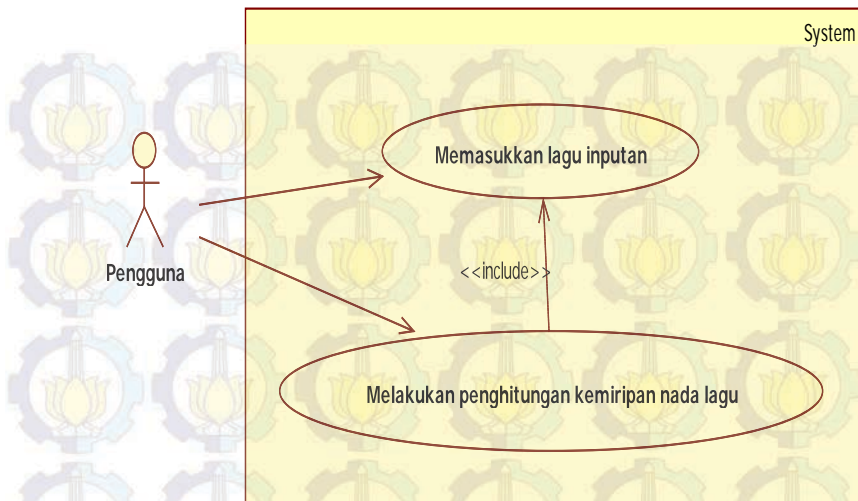
Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	Memasukkan lagu inputan	Pengguna dapat memasukkan lagu yang ingin dihitung kemiripannya dengan lagu yang ada pada <i>dataset</i>
F-0002	Melakukan penghitungan kemiripan nada lagu	Pengguna dapat melakukan penghitungan kemiripan nada lagu setelah memasukkan lagu inputan

3.1.4. Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Aktor yang terdapat pada sistem ini hanya memiliki sebuah peran yaitu sebagai pengguna. Pada sistem ini terdapat sebuah aktor yaitu pengguna aplikasi pendeteksi kemiripan nada lagu.

3.1.5. Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan digambarkan dalam tabel penjelasan kasus penggunaan dan diagram kasus penggunaan. Tabel penjelasan kasus penggunaan dapat dilihat pada Tabel 4-2 dan diagram kasus penggunaan dapat dilihat pada Gambar 3-2.



Gambar 3-2. Diagram Kasus Pengguna

Tabel 4-2. Daftar Kode Diagram Kasus Pengguna

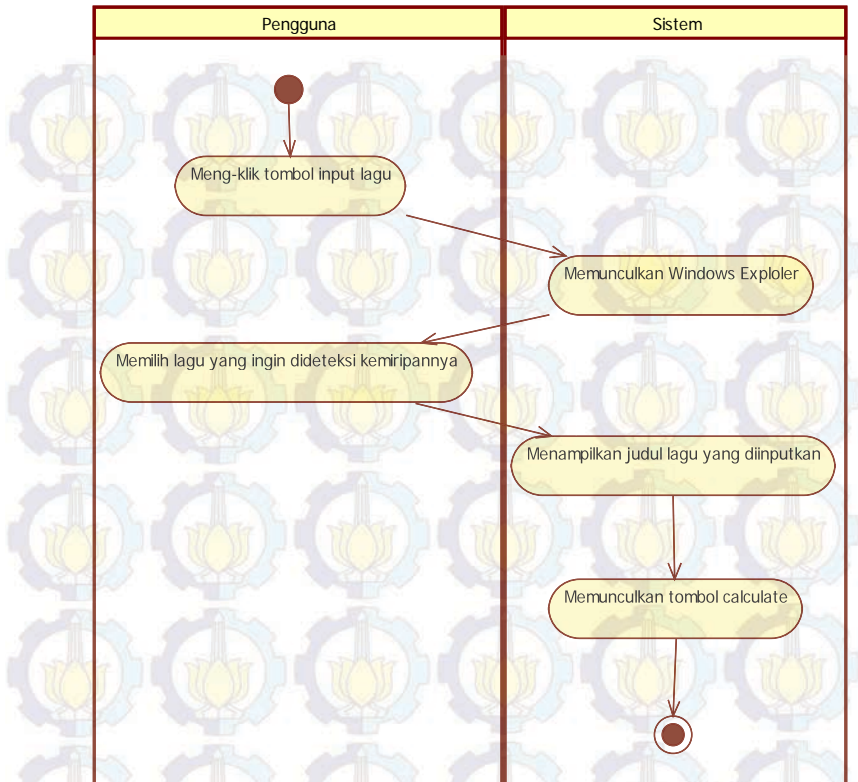
Kode Kasus Pengguna	Nama
UC-0001	Memasukkan lagu inputan
UC-0002	Melakukan peghitungan kemiripan nada lagu

3.1.5.1. Memasukkan Lagu Inputan (UC-0001)

Kasus penggunaan nomor UC-0001 adalah kasus penggunaan untuk memasukkan lagu inputan. Sebelum melakukan perhitungan kemiripan nada lagu, pengguna diharuskan memasukkan lagu yang ingin di hitung kemiripan nada lagunya. Untuk penjelasan lebih rinci dapat dilihat pada. menjelaskan mengenai diagram aktivitas dalam kasus pengguna memasukkan lagu inputan.

Tabel 4-3. Spesifikasi Kasus Penggunaan Memasukkan Lagu Inputan

Nama	Memasukkan lagu inputan
Kode	UC-0001
Deskripsi	Pengguna memasukkan lagu yang ingin digitung kemiripannya dengan lagu yang ada pada dataset
Tipe	Fungsional
Pemicu	Pengguna meng-klik tombol <i>input</i>
Aktor	Pengguna
Kondisi Awal	Tidak Ada
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna meng-klik tombol <i>input</i>. 2. Sistem memunculkan <i>windows exploler</i>. 3. Pengguna memilih lagu yang ingin dideteksi kemiripannya. 4. Sistem melakukan <i>loading</i> untuk lagu yang diinputkan. 5. Sistem menampilkan judul lagu inputan. 6. Sistem memunculkan tombol <i>calculate</i>. 7. Sistem menyembunyikan tombol <i>input</i>.
- Kejadian Alternatif	Tidak Ada
Kondisi Akhir	Sistem menampilkan judul lagu inputan dan memunculkan tombol <i>calculate</i>



Gambar 3-2. Diagram Aktivitas Kasus Penggunaan Memasukkan Lagu Inputan

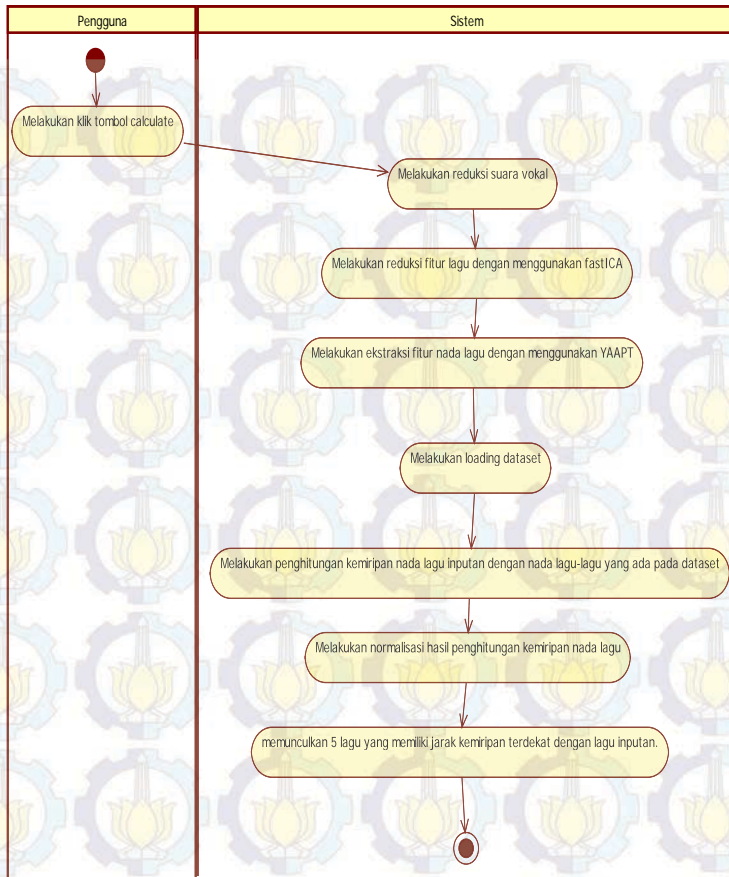
3.1.5.2. Melakukan Penghitungan Kemiripan Nada Lagu (UC-0002)

Kasus penggunaan nomor UC-0002 adalah kasus penggunaan untuk melakukan penghitungan kemiripan nada lagu inputan pengguna dengan nada lagu-lagu yang ada pada *dataset*. Untuk penjelasan lebih rinci dapat dilihat pada. menjelaskan

mengenai diagram aktivitas dalam kasus pengguna melakukan penghitungan kemiripan nada lagu.

Tabel 4-4. Spesifikasi Kasus Penggunaan Melakukan Penghitungan Kemiripan Nada Lagu

Nama	Melakukan penghitungan kemiripan nada lagu
Kode	UC-0002
Deskripsi	Sistem melakukan penghitungan kemiripan nada lagu inputan dengan nada lagu-lagu pada dataset
Tipe	Fungsional
Pemicu	Pengguna meng-klik tombol <i>calculate</i> yang muncul setelah melakukan input lagu
Aktor	Pengguna
Kondisi Awal	Lagu yang akan dicek telah diinputkan
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna melakukan klik tombol <i>calculate</i> 2. Sistem menyembunyikan tombol <i>calculate</i>. 3. Sistem melakukan reduksi suara vokal. 4. Sistem melakukan ekstraksi fitur nada lagu dengan menggunakan YAAPT. 5. Sistem akan melakukan re-dimensi nada lagu dengan menggunakan FFT 6. Sistem melakukan <i>loading dataset</i>. 7. Sistem melakukan penghitungan kemiripan nada lagu masukan dengan nada lagu-lagu yang ada pada <i>dataset</i>. 8. Melakukan normalisasi hasil penghitungan kemiripan 9. Sistem memunculkan 5 lagu yang memiliki jarak kemiripan terdekat dengan lagu inputan.
- Kejadian Alternatif	Tidak Ada
Kondisi Akhir	Sistem memunculkan 5 lagu yang memiliki jarak kemiripan terdekat dengan lagu inputan



Gambar 3-3. Diagram Aktivitas Kasus Penggunaan Melakukan Penghitungan Kemiripan Nada Lagu

3.2. Perancangan Sistem

Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu :

1. Pendeskripsian lingkungan perancangan perangkat lunak.

2. Perancangan metode secara umum.
3. *Preprocessing* untuk mereduksi suara vokal lagu inputan.
4. Pengimplementasian *Yet Another Algorithm for Pitch Tracking* dan re-dimensi nada lagu dengan menggunakan *Fast Fourier Transform*.
5. Pengimplementasian *Dynamic Time Warping* untuk penghitungan kemiripan nada lagu masukan dengan nada lagu-lagu yang ada pada *dataset*, sehingga dapat diambil 5 lagu yang paling mirip dengan lagu inputan berdasarkan nadanya.
6. Pengimplementasian normalisasi hasil dari penghitungan kemiripan nada lagu.
7. Perancangan antarmuka pengguna.

3.2.1. Lingkungan Perancangan Perangkat Lunak

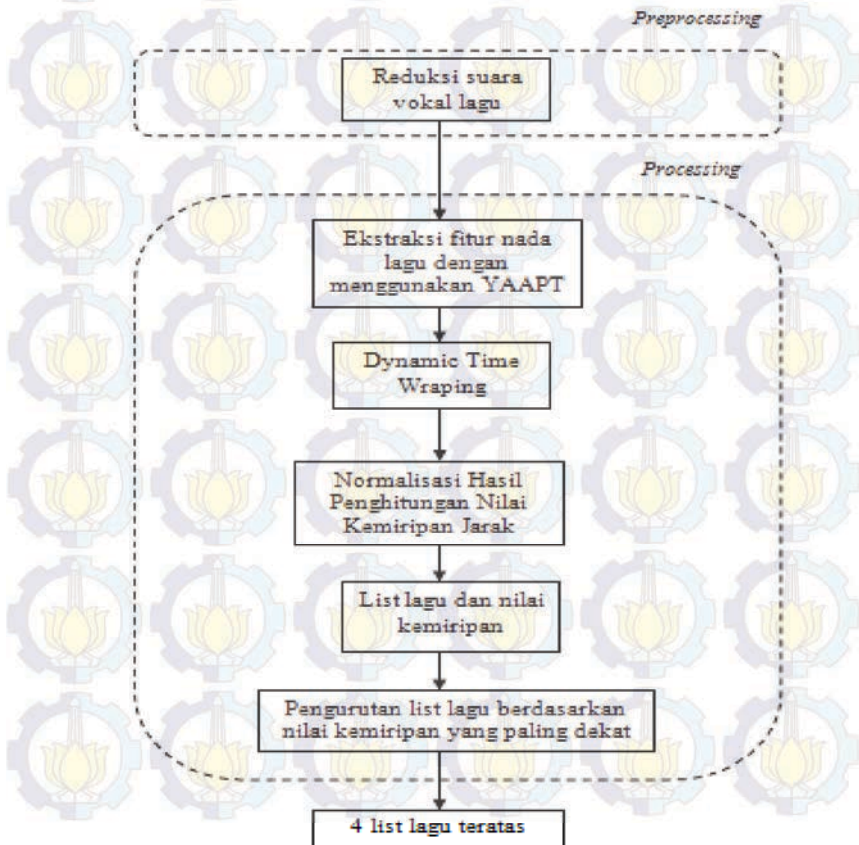
Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap perancangan perangkat lunak Tugas Akhir ini seperti dijelaskan pada Tabel 4-5.

Tabel 4-5 Lingkungan Perancangan Perangkat Lunak

Perangkat Keras	Prosesor	Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz (4 CPUs), ~2.2GHz
	Memori Primer	2 GB
	Memori Sekunder	2 GB
Perangkat Lunak	Sistem Operasi	Windows 7 Pro 64-bit
	Perangkat Lunak Utama	Matlab R2013a

3.2.2. Perancangan Metode Secara Umum

Metode ini hanya memiliki satu tahap inti, yaitu tahap penghitungan kemiripan nada lagu dengan menggunakan *Dynamic Time Warping*. Dimulai dari inputan berupa nada lagu hasil tahap *preprocessing*. Kemudian selanjutnya adalah tahap inti yaitu *Dynamic Time Warping* untuk melakukan penghitungan kemiripan nada lagu yang di-*input*-kan dengan nada lagu-lagu yang ada pada dataset. Diagram blok dari proses sistem ditunjukkan oleh Gambar 3-4.



Gambar 3-4. Diagram Blok Proses Sistem

3.2.3. Reduksi Suara Vokal Lagu

Pada sebagian besar perekaman lagu, *track* vokal memiliki nilai yang berdekatan pada kanal kiri maupun kanan. *Volume* yang dihasilkan oleh instrumen-instrumen lagu didistribusikan secara merata diantara 2 kanal. Dimana vokal suara disisipkan pada nilai yang sama pada kedua kanal. Yang dimana akan terjadi pereduksian suara apabila dilakukan operasi pengurangan antara 2 kanal tersebut. *Pseudocode* dapat dilihat pada Gambar 3-5.

Masukan	Matriks lagu inputan
Keluaran	Matriks yang berisi lagu yang vokalnya telah tereduksi
1. Mengambil nilai matriks lagu inputan sebelah kiri audio	
2. Mengambil nilai matriks lagu inputan sebelah kanan audio	
3. Mengurangkan nilai matriks lagu inputan sebelah kiri dengan sebelah kanan	
4. Mengurangkan nilai matriks lagu inputan sebelah kanan dengan sebelah kiri	
5. Menyimpan kedua nilai hasil pengurangan	

Gambar 3-5. *Pseudocode* Reduksi Suara Vokal Lagu

Vokal pada lagu akan masih terdengar karena lagu yang diinputkan menggunakan efek *revibrasi stereo* yang menyebabkan menggemanya vokal masuk dan keluar antara kanal kiri dan kanan.

3.2.4. Eksraksi Fitur Nada Lagu

Sebuah lagu memiliki banyak fitur yang ada di dalamnya, salah satunya adalah nada lagu. Untuk mendapatkan nada lagu pada sebuah lagu maka diperlukan sebuah algoritma untuk menemukan fitur nada pada suatu lagu. *Yet Another Algorithm for Pitch Tracking* (YAAPT) merupakan salah satu algoritma untuk

menemukan fitur nada pada suatu lagu. YAAPT menggunakan spektografik lagu untuk menentukan *pitch- pitch* pada lagu yang akan disusun menjadi deretan yang disebut dengan nada lagu.

Pengerjaan ekstraksi fitur nada lagu menggunakan *library* YAAPT, dimana masukan dalam proses ini adalah lagu hasil reduksi suara vokalnya yang diambil kanal sebelah kiri ataupun kanannya saja. Hal ini dikarenakan YAAPT hanya bisa memproses lagu dengan 1 dimensi saja. YAAPT juga meminta masukan apakah lagu memiliki suara vokal ataupun tidak, dan pada pengerjaan ini, dianggap lagu memiliki suara vokal serta tidak memiliki parameter lebih lainnya. *Pseudocode* YAAPT dapat dilihat pada Gambar 3-6.

Masukan	Matriks data lagu yang telah direduksi suara vokalnya
Keluaran	Matriks hasil ekstraksi fitur nada lagu dan jumlah frame yang telah dihitung
1. Mengambil matriks data lagu yang telah direduksi vokalnya, dan diambil sisi kirinya saja. 2. Melakukan pemanggilan fungsi YAAPT untuk ekstraksi fitur nada.	

Gambar 3-6. Pseudocode Ekstraksi Fitur Nada Lagu

Variabel *Pitch* menunjukkan matrik hasil ekstraksi fitur nada lagu, sedangkan *nf* menunculkan jumlah frame yang telah dihitung.

3.2.5. Re-Dimensi Atribut

Data sebuah lagu memiliki dimensi fitur yang sangat besar. Besarnya dimensi suatu fitur tentu akan sangat mempengaruhi kecepatan dalam proses pengolahan data. Oleh karena itu perlu dilakukan re-dimensi fitur. *Fast Fourier Transform* algoritma untuk menghitung transformasi *fourier* diskrit dengan cepat dan efisien. Karena banyak sinyal-sinyal

dalam sistem komunikasi yang bersifat kontinyu, sehingga untuk kasus sinyal kontinyu kita gunakan transformasi fourier.

Pengerjaan re-dimensi ini menggunakan *tools* yang telah disediakan oleh matlab, dimana masukan dalam proses ini adalah lagu yang telah direduksi vokalnya dan telah melewati proses ekstraksi nada dengan menggunakan YAAPT. Akan dilakukan penyusutan dimensi, sehingga dimensinya hanya 1/11 kali dari dimensi awalnya, dan kemudian kan dikembalikan lagi ke bentuk awalnya dengan menggunakan *inverse FFT*. *Pseudocode FFT* dapat dilihat pada Gambar 3-7.

Masukan	Matriks data nada lagu inputan
Keluaran	Data yang telah dire-dimensi menggunakan FFT
1. Mengambil matriks lagu yang telah diekstraksi nadanya 2. Melakukan fft 3. Melakukan redimensi menjadi 1/11 panjang matriks nada lagu 4. Melakukan ifft terhadap matriks nada lagu yang telah dire-dimensi	

Gambar 3-7. Pseudocode Re-dimensi Atribut

Variabel hasil menunjukkan hasil matriks yang telah dire-dimensi.

3.2.6. *Dynamic Time Warping*

Untuk menentukan kemiripan nada lagu, maka diperlukan sebuah metode untuk menghitung nilai kemiripan sebuah lagu. *Dynamic Time Warping* merupakan algoritma yang menghitung *optimal warping path* antara dua waktu. Algoritma ini menghitung baik antara nilai *warping path* dari dua waktu dan jaraknya.

Pengerjaan penghitungan jarak kemiripan nada lagu menggunakan algoritma dtw, dimana masukan dalam proses ini adalah nada lagu inputan serta nada lagu-lagu yang ada pada

dataset. *Pseudocode Dynamic Time Warping* dapat dilihat pada Gambar 3-8.

Variabel *fft11* adalah hasil dari pengambilan data dari dataset nada lagu yang disiapkan, yang akan di bandingkan dengan nada lagu yang dimasukkan. Penghitungan jarak ini dilakukan secara rekursif sejumlah lagu yang ada pada dataset.

Masukan	Matriks data nada lagu inputan yang telah di re-dimensi dan nada lagu-lagu yang ada pada dataset
Keluaran	Matriks judul lagu dataset dan nilai kedekatannya dengan lagu yang diinputkan
<ol style="list-style-type: none"> 1. Mengambil matriks nada lagu yang telah di re-dimensi 2. <code>load('fft11.mat')</code> 3. Pengambilan nama-nama variabel pada <code>fft11.mat</code> 4. <code>nsong = jumlah lagu yang ada pada fft11.mat</code> 5. <code>for n = 1 : nsong</code> 6. Melakukan pemanggilan fungsi <i>Dynamic Time Warping</i> dengan masukan matriks nada lagu yang telah dire-dimensi dan nada lagu pada <code>fft11(n)</code>. 7. Melakukan penyimpanan nama variabel pada <code>fft11.mat</code> yang telah di hitung kemiripannya dengan nada lagu masukan 8. Melakukan penyimpanan nilai jarak dari seluruh nada pada <code>fft11.mat</code> yang telah di hitung kemiripannya dengan nada lagu masukan 9. <code>end for</code> 10. <code>JarakKeseluruhan = Hasil seluruh penghitungan jarak kemiripan nada lagu</code> 	

Gambar 3-8. *Pseudocode Dynamic Time Warping*

Hasil dari perhitungan ini adalah variabel rank yang berisi judul lagu serta nilai kemiripannya dengan lagu inputan, yang selanjutnya akan menghasilkan variabel JarakKeseluruhan.

3.2.7. Normalisasi Hasil *Dynamic Time Warping*

Hasil dari perhitungan kemiripan nada lagu cenderung besar, dan kurang sesuai dalam penampakkannya. Maka untuk menanggulanginya diperlukan suatu metode normalisasi.

Pengerjaan normalisasi hasil perhitungan kemiripan nada lagu dilakukan dengan cara membagi nilai yang ada dengan nilai rata-rata dari semua data jarak kemiripan nada lagu yang telah dihitung.

Pseudocode Dynamic Time Warping dapat dilihat pada Gambar 3-9.

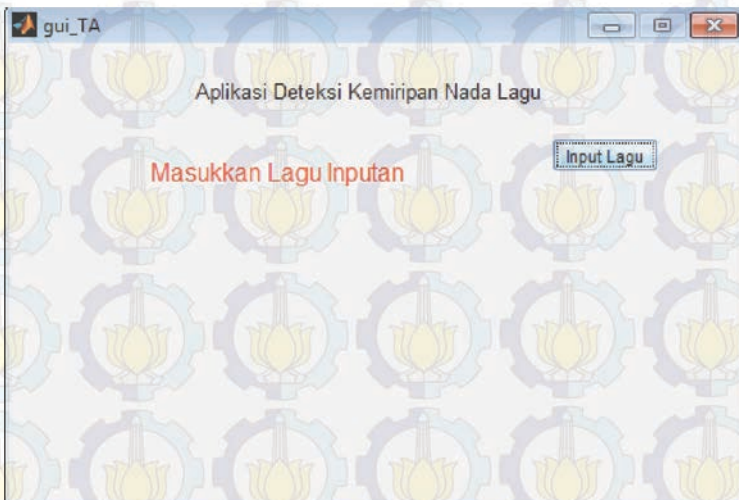
Masukan	Matriks data jarak kemiripan nada lagu
Keluaran	Data jarak kemiripan yang telah dinormalisasi dan telah diurutkan dari jarak terpendek
1. Mengambil matriks data jarak kemiripan nada lagu 2. Melakukan penghitungan nilai rata-rata jarak kemiripan pada matriks 3. For n=1 : nsong 4. Melakukan pembagian nilai jarak kemiripan dengan nilai rata-rata jarak kemiripan 5. End for 6. RankingFin = nama tiap variabel judul yang ada pada fft11.mat beserta nilai jarak kemiripan nada lagunya yang telah dinormalisasi dan diurutkan dari yang terkecil nilainya	

Gambar 3-9. *Pseudocode Normalisasi Hasil Dynamic Time Warping*

3.2.8. Perancangan Antarmuka Pengguna

Bagian ini membahas rancangan tampilan antar muka pada sistem. Pada sistem ini terdapat satu tampilan utama yaitu kanvas pada saat pertama kali aplikasi dijalankan.

Halaman ini pada merupakan tampilan utama yang muncul ketika sistem pertama kali dijalankan. Pada halaman ini terdapat menu Input Lagu. Secara default, sewaktu pertama kali sistem dijalankan, tidak ada lagu yang diinputkan, sehingga hanya tombol Input Lagu yang dimunculkan. Apabila pengguna telah memasukkan lagu, maka akan muncul tombol *calculate* seperti pada Gambar 3-11. Penjelasan detil atribut antarmuka dijelaskan pada Tabel 4-6. Sedangkan contoh tampilan akhir saat proses penghitungan kemiripan akan ditunjukkan pada Gambar 3-12.



Gambar 3-10. Tampilan Aplikasi Deteksi Kemiripan Nada Lagu

Tabel 4-6. Spesifikasi Atribut Antar Muka Tampilan Utama

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	Input Lagu	Button	Memasukkan lagu yang ingin di hitung kemiripannya	Action
2	Calculate	Button	Melakukan proses penghitungan kemiripan	Action

**Gambar 3-11. Tampilan Aplikasi Setelah Input Lagu**



Gambar 3-12. Tampilan Akhir Setelah Proses Perhitungan Kemiripan

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1. *Lingkungan Implementasi*

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah *Matlab R2013a* yang diinstal pada sistem operasi *Windows 7*.

4.2. *Implementasi*

Pada subbab ini akan dijelaskan implementasi setiap subbab yang terdapat pada bab sebelumnya yaitu bab perancangan program. Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

4.2.1. Implementasi Fungsi `wavRead()`

Fungsi `wavRead()` adalah fungsi untuk membaca data ber ekstensi `.wav` dari yang kemudian akan disimpan dalam variabel yang berupa matrix. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4.1.

Fungsi utama untuk membaca file dengan ekstensi `.wav` dari windows terdapat pada baris 1 sampai 3. Baris 4 sampai 7 menjelaskan tentang penyimpanan alamat file `.wav` pada *windows*, matriks lagu, serta frekuensi sampel lagu yang diinputkan untuk disimpan dalam *handles* aplikasi, agar dapat diakses oleh fungsi-fungsi yang lainnya.

1	[filename pathname] = uigetfile({'*.wav'}, 'Select File');
2	fullpathname = strcat (pathname, filename);
3	[audio,fs] = wavread(fullpathname);
4	handles.fullpathname = fullpathname;
5	handles.audio = audio;
6	handles.fs = fs;
7	guidata(hObject, handles);

Kode Sumber 4.1. Kode Sumber Fungsi wavRead (1)

4.2.2. Implementasi Fungsi Reduksi Suara Vokal Lagu

Fungsi reduksi suara vokal lagu digunakan untuk mengurangi suara vokal penyanyi, karena pada aplikasi yang dibangun ini mengutamakan suara nada lagu, bukan suara penyanyi. Implementasi fungsi reduksi suara vokal lagu dapat dilihat pada Kode Sumber 4.2.

1	left1=handles.audio(:,1);
2	right1=handles.audio(:,2);
3	take1 = left1 - right1;
4	take2 = right1 - left1;
5	takefin(:,1) = take1;

Kode Sumber 4.2 Kode Sumber Reduksi Suara Vokal Lagu

Matriks hasil pembacaan lagu berekstensi .wav yang memiliki tipe *stereo* akan memiliki matrix $2 \times N$, sehingga salah satu cara untuk mereduksi suara vokal pada lagu yaitu dengan cara mengurangi kedua sisi matriks satu sama lain dan mengambil hasilnya 1 sisi saja.

4.2.3. Implementasi Fungsi YAAPT()

Fungsi **YAAPT()** digunakan sebagai metode peng-ekstraksian dari sebuah lagu menjadi deretan nada lagu. Fungsi YAAPT sendiri, dibuat oleh Speech Communication Laboratory pada State University of New York yang bertempat di Binghampton. Sehingga, pada tugas akhir ini hanya akan memanggil fungsinya saja untuk dimplementasikan. Pengimplementasiannya dapat dilihat pada Kode Sumber 4.3.

1	<code>[Pitch, nf] = yaapt(takefin, handles.fs, 1, []);</code>
2	<code>Pitch_fixed = ptch_fix(Pitch);</code>
3	<code>Pitch_fixed =transpose(Pitch_fixed);</code>

Kode Sumber 4.3. Kode Sumber Implementasi Fungsi YAAPT

Pemanggilan fungsi YAAPT sendiri membutuhkan beberapa parameter, `takefin` menunjukkan data lagu yang akan diproses, `handles.fs` merupakan frekuensi sampel dari lagu yang diinputkan, konstanta 1 menunjukkan bahwa lagu yang diinputkan memiliki suara dan 0 bila tidak ada suaranya, parameter `[]` menunjukkan bahwa tidak ada parameter tambahan lagi untuk proses ekstraksi nada lagu. YAAPT juga memiliki fungsi untuk membetulkan hasil dari ekstraksi fitur, yaitu dengan cara memanggil fungsi `ptch_fix()` yang ditunjukkan pada baris 2.

4.2.4. Implementasi Fungsi `fft()`

Pengerjaan re-dimensi ini menggunakan *tools* yang telah disediakan oleh matlab, dimana masukan dalam proses ini adalah lagu yang telah direduksi vokalnya dan telah melewati proses ekstraksi nada dengan menggunakan YAAPT. Akan dilakukan penyusutan dimensi, sehingga dimensinya hanya 1/11 kali dari dimensi awalnya, dan kemudian kan dikembalikan lagi ke bentuk awalnya dengan menggunakan *inverse FFT*. Pengimplementasiannya dapat dilihat pada Kode Sumber 4.4.

1	fa = fft(Pitch_fixed);
2	fb = fa(1:length(fa)/11);
3	hasil = real(ifft(fb));

Kode Sumber 4.4. Kode Sumber Fungsi FFT()

4.2.5. Implementasi Fungsi dtw()

Fungsi **dtw** () digunakan untuk menghitung nilai kemiripan nada lagu inputan dengan nada-nada lagu yang ada pada dataset. Pada fungsi ini, akan diimplementasikan algoritma *Dynamic Time Warping* dalam perhitungan kemiripannya. Proses tersebut diimplementasikan pada Kode Sumber 4.5 dan Kode Sumber 4.6.

Kode Sumber Fungsi dtw() (1) menunjukkan cara untuk memanggil fungsi **dtw**() dengan yang membutuhkan 2 parameter, yaitu 2 data yang akan di uji kemiripannya. Fungsi **dtw**() diimplementasikan berdasarkan algoritma *dynamic time warping* yang telah dijelaskan pada bab sebelumnya.

1	load('fft11.mat');
2	fields = fieldnames(fft11);
3	nsong=length(fields);
4	for n = 1 : nsong
5	d = dtw(hasil,fft11.(fields{n}));
6	c(n) = d;
7	clear d;
8	end

Kode Sumber 4.5. Kode Sumber Fungsi dtw (1)

4.2.6. Implementasi Fungsi Normalisasi Hasil *Dynamic Time Warping*

Implementasi fungsi normalisasi hasil *Dynamic Time Warping* digunakan untuk menormalkan hasil dari penghitungan kemiripan nada lagu yang hasilnya sangat besar. Implementasi

fungsi normalisasi hasil *Dynamic Time Warping* dapat dilihat pada Kode Sumber 4.7

1	<code>function d=dtw(s,t,w)</code>
2	<code>f nargin<3</code>
3	<code>w=Inf;</code>
4	<code>end</code>
5	<code>ns=size(s,1);</code>
6	<code>nt=size(t,1);</code>
7	<code>if size(s,2)~=size(t,2)</code>
8	<code>error('Error in dtw(): the dimensions of the two input signals do not match.');</code>
9	<code>end</code>
10	<code>w=max(w, abs(ns-nt));</code>
11	<code>D=zeros(ns+1,nt+1)+Inf;</code>
12	<code>D(1,1)=0;</code>
13	<code>for i=1:ns</code>
14	<code>for j=max(i-w,1):min(i+w,nt)</code>
15	<code>oost=norm(s(i,:)-t(j,:));</code>
16	<code>D(i+1,j+1)=oost+min([D(i,j+1), D(i+1,j), D(i,j)]);</code>
17	<code>end</code>
18	<code>end</code>
19	<code>d=D(ns+1,nt+1);</code>

Kode Sumber 4.6. Kode Sumber Fungsi dtw (2)

1	<code>maximum = max(c);</code>
2	<code>for i=1 :length(c)</code>
3	<code>normalized(i) = c(i)/maximum;</code>
4	<code>end;</code>
5	<code>rank(g,1) = fields(g);</code>
6	<code>rank{g,2} = num2str(normalized(g));</code>

Kode Sumber 4.7. Kode Sumber Fungsi Normalisasi

Kode Sumber Fungsi Normalisasi Hasil *Dynamic Time Warping* melakukan normalisasi terhadap data hasil perhitungan

ynag dilakukan pada fungsi `dtw()`. Kemudian hasil dari normalisasi tersebut akan disimpan pada variabel *rank*.

4.2.7. Implementasi Fungsi *Ranking* Nada Lagu

Implementasi fungsi *ranking* nada lagu digunakan untuk menemukan urutan kemiripan nada-nada lagu yang ada pada dataset dengan nada lagu inputan. Hasil dari pemeringkatan ini akan diambil 5 teratas dan akan ditampilkan pada GUI. Implementasi fungsi *ranking* nada lagu dapat dilihat pada Kode Sumber 4.8

Kode Sumber Fungsi *Ranking* Lagu melakukan pengurutan terhadap variable *RankingFin* yang didapatkan dari fungsi `dtw()`, yang berisi judul lagu pada dataset dan jaraknya dengan lagu inputan. Pengurutannya berdasarkan nilai jarak terpendek antar nada lagu yang ada pada dataset dengan nada lagu inputan. Kemudian data 5 teratas akan ditampilkan pada GUI.

1	<code>RankingFin = sortrows(rank,2);</code>
2	<code>set(findobj('Tag','text4'),'String',RankingFin{1});</code>
3	<code>set(findobj('Tag','text5'),'String',RankingFin{2});</code>
4	<code>set(findobj('Tag','text6'),'String',RankingFin{3});</code>
5	<code>set(findobj('Tag','text7'),'String',RankingFin{4});</code>
6	<code>set(findobj('Tag','text16'),'String',RankingFin{5});</code>
7	<code>set(findobj('Tag','text4'),'Visible','On');</code>
8	<code>set(findobj('Tag','text5'),'Visible','On');</code>
9	<code>set(findobj('Tag','text6'),'Visible','On');</code>
10	<code>set(findobj('Tag','text7'),'Visible','On');</code>
11	<code>set(findobj('Tag','text16'),'Visible','On');</code>

Kode Sumber 4.8. Kode Sumber Fungsi *Ranking* Nada Lagu

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan tugas akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor : Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz (4 CPUs), ~2.2GHz

Memori : 4.00 GB

Jenis Device : Laptop

Sistem Operasi : Microsoft Windows 7 Pro 64 bit

5.2. Dataset dan Data Testing

Dataset dan *data testing* yang akan digunakan merupakan data-data lagu yang diambil dari internet. Jumlah *dataset* yang disiapkan berjumlah 20 lagu, dimana di dalam 20 lagu tersebut ada 12 lagu yang digunakan untuk testing plagiarisme, dan sisanya adalah *noise* untuk pembandingan hasil perhitungan kemiripan. Semua *dataset* yang dimasukkan telah melewati proses pengurangan suara vokal lagu, ekstraksi nada menggunakan YAAPT serta telah melalui proses *fft* seperti yang telah dijelaskan pada subbab 3.2. *Dataset* tersebut juga akan digunakan 3 lagu untuk pengujian fungsionalitas dari metode *dynamic time*

warping, serta akan digunakan 5 lagu untuk pengujian pendeteksian lagu yang dinyanyikan bukan oleh penyanyi aslinya. Untuk detail data lagu yang dijadikan *dataset* akan dijelaskan pada Tabel 5-1 dan Tabel 5-2 dan untuk detail data uji akan dijelaskan pada Tabel 5-3 dan Tabel 5-4.

Tabel 5-1. Tabel Data Lagu (1)

Data Lagu		
Judul Lagu	Nama Penyanyi	Durasi Lagu
Because Of You	Kelly Clarkson	00:03:37
Proud Of You	Fiona Fung	00:03:10
You Are Still The One	Shania Twain	00:03:34
Goodbye My Lover	James Blunt	00:04:18
Need You Now	Lady Antebellum	00:03:55
Sweet Little Sixteen	Chuckberry	00:03:11
Lust For Life	Iggy Pop	00:03:12
Underpressure	Queen	00:04:10
Creep	Radiohead	00:03:55
He's So Fine	The Ciffons	00:01:55
I Want A New Drug	Huey Lewis and The News	00:03:29
There's No Tomorrow	Tony Martin	00:03:19
Do Somethin	Britney Spears	00:03:32
Close	Westlife	00:04:04
Moliendo Cafe	Julio Iglesias	00:03:52
I'll Be There	Jackson 5	00:03:50
Time Is Running Out	Muse	00:04:04
Makes Me Wonder	Maroon 5	00:03:42

Tabel 5-2. Tabel Data Lagu (2)

Data Lagu		
Judul Lagu	Nama Penyanyi	Durasi Lagu
Picture Book	The Kinks	00:02:39
All About That Bass	Meghan Trainor	00:03:09

Tabel 5-3. Tabel Data Uji Lagu (1)

Data Uji		
Judul Lagu	Nama Penyanyi	Durasi Lagu
Are You Gonna Be My Girl	Jet	00:03:34
It's Now or Never	Elvis Presley	00:03:19
'O Sole Mio	Luciano Pavarotti	00:03:21
The Air That I Breathe	The Hollies	00:04:08
Pusing Pala Berbie	Putri Bahar	00:03:48
Kopi Dangdut	The Sallys	00:03:45
Bersamamu	Vierra	00:04:15
Surat Undangan	Diah Iskandar	00:03:00
Pejuang Cinta	Hello	00:03:45
Mahluk Tuhan Paling Sexy	Mulan Jameela	00:04:28
Proud Of You (Cover)	Jane Callista	00:04:07

Tabel 5-4. Tabel Data Uji Lagu (2)

Datatest		
Judul Lagu	Nama Penyanyi	Durasi Lagu
I'll Be There (Cover)	Westlife	00:03:55
Goodbye My Lover (Cover)	Eliseo	00:04:50
Close (Cover)	LadeeMusicify(Youtube)	00:03:41
You're Still The One (Cover)	Lee Dewyze	00:03:05
Surfin USA	Beach Boys	00:03:40
Ghostbusters	Ray Parker Jr.	00:04:04
Lust For Life	Iggy Pop	00:03:12
Do Somethin	Britney Spears	00:03:32
Underpressure	Queen	00:04:10

5.3. Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas. Pengujian fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada kesesuaian hasil keluaran sistem. Serta akan dilakukan pengujian akurasi dari sistem untuk mengetahui kesesuaian hasil keluaran sistem apakah sesuai dengan tujuan dari dibuatnya sistem atau tidak.

Pengujian terhadap hasil perhitungan kemiripan nada lagu dilakukan dengan membandingkan 5 hasil teratas perhitungan kemiripan nada lagu sistem dengan hasil kemiripan nada lagu yang seharusnya dihasilkan berdasarkan data yang didapatkan

melalui internet. Apabila salah satu dari 5 hasil teratas mengeluarkan judul lagu yang mirip dengan tepat, maka sistem dianggap telah berhasil melakukan perhitungan kemiripan nada lagu. Setelah dilakukan uji coba maka akan dilakukan penghitungan akurasi seperti yang ditunjukkan pada rumus 5-1.

$$Akurasi = \frac{Jumlah\ Uji\ Coba\ Yang\ Berhasil}{Jumlah\ Uji\ Coba} \times 100\% \quad (5-1)$$

Uji coba ini dilakukan untuk menguji apakah fungsionalitas program telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya. Uji coba akan didasarkan pada beberapa skenario untuk menguji kesesuaian dan kinerja aplikasi.

Pengujian terdiri dari 3 skenario yaitu:

1. Pengujian fungsionalitas *Dynamic Time Warping*, pengujian ini bertujuan untuk mengecek apakah fungsi *Dynamic Time Warping* sudah diimplementasikan dengan benar atau belum.
2. Pengujian penghitungan kemiripan dengan inputan lagu *cover*, bertujuan untuk melihat apakah aplikasi yang dibangun dapat melakukan penghitungan kemiripan nada lagu terhadap lagu *cover* salah satu lagu yang ada pada *dataset* dengan baik.
3. Pengujian penghitungan kemiripan dengan inputan lagu yang diopinikan melakukan plagiarisasi terhadap salah satu lagu pada *dataset*, bertujuan untuk melihat apakah aplikasi yang dibangun dapat melakukan penghitungan kemiripan nada lagu terhadap inputan lagu yang diopinikan melakukan plagiarisasi terhadap salah satu lagu yang ada pada *dataset* dengan baik.

Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolak ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.5. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

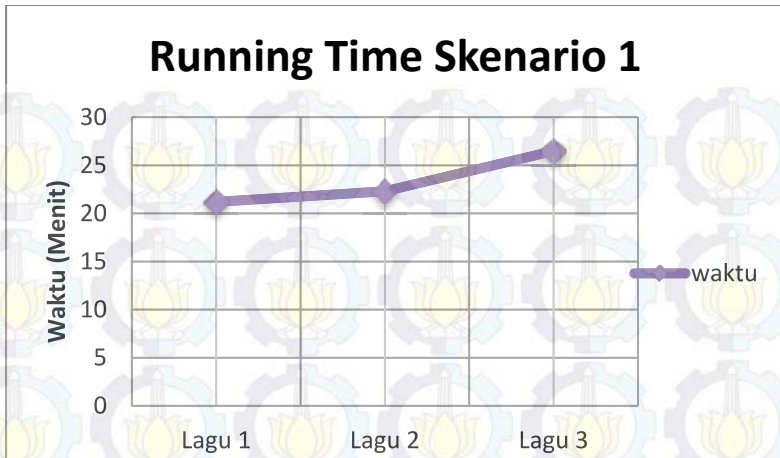
5.3.1. Pengujian Fungsionalitas *Dynamic Time Warping*

Pengujian fungsionalitas *Dynamic Time Warping* bertujuan untuk mengecek apakah fungsi *dynamic time warping* untuk perhitungan kemiripan berjalan dengan benar. Skenarionya adalah, dengan cara menginputkan salah satu lagu yang ada pada dataset. Apabila kemiripan lagu inputan dengan lagu yang sama pada dataset memiliki kemiripan dengan jarak 0, maka pengujian fungsionalitas *dynamic time warping* dianggap telah berhasil. Skenario pertama ini dilakukan sebanyak 3 kali.. Hasil uji dari skenario pertama ini dapat dilihat Tabel 5-5. Sedangkan untuk *screenshot* hasil uji coba skenario 1 dapat dilihat pada bagian lampiran.

Ketiga ujicoba tersebut menunjukkan hasil yang valid, dimana apabila kita memasukkan salah satu lagu yang sama dengan lagu yang ada pada dataset maka lagu inputan dan lagu yang sama pada data set akan memiliki nilai jarak 0. Sehingga dapat disimpulkan bahwa pengimplementasian perhitungan kemiripan menggunakan *dynamic time warping* telah dilakukan dengan benar Untuk menentukan kesesuaian keluaran dengan program dan analisa yang telah dibuat diperlukan pengujian akurasi dan *running time*. Perhitungan akurasi dilakukan menggunakan rumus 5-1 Dari hasil penghitungan akurasi terhadap skenario maka didapatkan akurasi sebesar 100%. Grafik hasil *running time* skenario pertama dapat dilihat pada Grafik 5-1.

Tabel 5-5. Tabel Hasil Uji Coba Skenario 1

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Lust For Life	1. Lust For Life	0
	2. Moliendo Cafe	0.1351
	3. Makes Me Wonder	0.1856
	4. Creep	0.19723
	5. Time Is Running Out	0.2124
Do Somethin	1. Do Somethin	0
	2. Hes So Fine	0.1351
	3. Sweet Little Sixteen	0.1856
	4. Moliendo Cafe	0.19723
	5. Creep	0.2124
Underpressure	1. Underpressure	0
	2. There's No Tomorrow	0.082923
	3. Need You Now	0.11946
	4. Close	0.1667
	5. You Are Still The One	0.1925



Grafik 5-1. Grafik Running Time Skenario 1

5.3.2. Pengujian Penghitungan Kemiripan Dengan Inputan Lagu Cover Dari Dataset

Pengujian penghitungan kemiripan dengan inputan lagu cover dari dataset bertujuan untuk mengecek apakah sistem yang dibangun dapat melakukan penghitungan kemiripan nada lagu yang ada pada dataset apabila lagu tersebut dinyanyikan oleh penyanyi yang berbeda meskipun lagu yang dibawakan sama, yang dibawakan tersebut telah diaransemen sebelumnya. Apabila salah satu diantara 5 lagu yang dimunculkan berhasil memunculkan judul lagu asli dari lagu inputan, maka dianggap berhasil. Hasil pemringkatan lagu didapatkan dari pemeringkatan hasil perhitungan jarak kemiripan menggunakan *Dynamic Time Warping*. Pengujian pada skenario kedua ini dilakukan sebanyak 5 kali. Hasil ujicoba dari skenario kedua ini ditunjukkan pada Tabel 5-6 dan Tabel 5-7. Sedangkan untuk *screenshot* hasil uji coba skenario 2 dapat dilihat pada bagian lampiran.

Tabel 5-6. Tabel Hasil Uji Coba Skenario 2 (1)

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Proud Of You (Cover)	1. Proud Of You	0.24113
	2. You Are Still The One	0.26611
	3. All About That Bass	0.26757
	4. Goodbye My Lover	0.26873
	5. I'll Be There	0.28426
I'll Be There (Cover)	1. Need You Now	0.25087
	2. There's No Tomorrow	0.25828
	3. Close	0.26976
	4. Time Is Running Out	0.3026
	5. Underpressure	0.31103
Goodbye My Lover (Cover)	1. Picture Book	0.00046649
	2. There's No Tomorrow	0.16842
	3. You Are Still The One	0.18163
	4. Need You Now	0.2561
	5. Proud Of You	0.28974

Tabel 5-7. Tabel Hasil Uji Coba Skenario 2 (2)

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Close (Cover)	1. Close	0.34085
	2. Time Is Running Out	0.34569
	3. Makes Me Wonder	0.35234
	4. Theres No Tommorow	0.39086
	5. Lust For Life	0.39854
You're Still The One (Cover)	1. Time Is Running Out	0.35133
	2. Makes Me Wonder	0.35664
	3. Lust For Life	0.39828
	4. Moliendo Cafe	0.40205
	5. Need You Now	0.44956

Untuk menentukan kesesuaian keluaran dengan program dan analisa yang telah dibuat diperlukan pengujian akurasi dan *running time*. Perhitungan akurasi dilakukan menggunakan rumus 5-1. Dari hasil kelima uji coba maka didapatkan akurasi sebesar 40% karena hanya 2 dari 5 percobaan yang berhasil dilakukan dengan benar.

Rendahnya akurasi ini dipengaruhi oleh aransemen pada nada lagu *cover* terhadap nada lagu aslinya, seperti pengubahan penggunaan alat musik, pengubahan tempo, dan lain-lain. Selain itu, rendahnya akurasi juga dapat dipengaruhi oleh nada dasar dari penyanyi, yang menyebabkan perbedaan permainan nada lagu

cover dengan nada lagu aslinya. Kedua hal tersebut menyebabkan pendeteksian menjadi tidak akurat. Grafik hasil *running time* skenario kedua dapat dilihat pada Grafik 5-2.



Grafik 5-2. Grafik Running Time Skenario 2

5.3.3. Pengujian Penghitungan Kemiripan Dengan Inputan Lagu Yang Diopinikan Melakukan Plagiarisasi Terhadap Salah Satu Lagu Pada Dataset

Pengujian penghitungan kemiripan dengan inputan lagu yang diopinikan melakukan plagiarisasi terhadap salah satu lagu pada dataset bertujuan untuk mengecek apakah sistem yang dibuat dapat bekerja sesuai dengan manfaat sistem yang diharapkan yaitu dapat membantu dalam mendeteksi plagiarisasi terhadap nada lagu. Pengujian pada skenario ketiga ini dilakukan sebanyak 12 kali. Apabila salah satu diantara 5 lagu yang dimunculkan berhasil memunculkan judul lagu asli dari lagu yang diopinikan melakukan plagiarisasi, maka dianggap berhasil. Hasil pemringkatan lagu didapatkan dari pemeringkatan hasil

perhitungan jarak kemiripan menggunakan *Dynamic Time Warping*. Daftar lagu yang diopinikan melakukan plagiarisasi akan dijelaskan pada Tabel 5-8. Hasil ujicoba 1 sampai 5 dari skenario ketiga ini ditunjukkan pada Tabel 5-9 , Tabel 5-10, Tabel 5-11, dan Tabel 5-12. Sedangkan untuk *screenshot* hasil dari uji coba skenario 3 dapat dilihat pada bagian lampiran.

Tabel 5-8. Tabel Lagu Yang Diopinikan Melakukan Plagiarisme

Tabel Lagu Yang Diopinikan Plagiat	
Judul Lagu Asal	Judul Diopinikan Plagiat
Lust For Life	Are You Gonna Be My Girl
There's No Tomorrow	Its Now Or Never
There's No Tomorrow	'O Sole Mio
Creep	The Air That I Breath
All About That Bass	Pusing Pala Berbie
Moliendo Cafe	Kopi Dangdut
Close	Bersamamu
I'll Be There	Surat Undangan
Makes Me Wonder	Pejuang Cinta
Time Is Running Out	Mahluk Tuhan Paling Sexy
Sweet Little Sixteen	Surfin USA
I Want A New Drug	Ghostbusters

Tabel 5-9. Tabel Hasil Uji Coba Skenario 3 (1)

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Are You Gonna Be My Girl	1. Lust For Life	0.15331
	2. Moliendo Café	0.15551
	3. Makes Me Wonder	0.17403
	4. Time Is Running Out	0.19352
	5. Creep	0.20856
It's Now Or Never	1. There's No Tomorrow	0.19269
	2. You Are Still The One	0.20756
	3. Need You Now	0.29328
	4. Proud Of You	0.33129
	5. Underpressure	0.35986
'O Sole Mio	1. You Are Still The One	0.24588
	2. There's No Tomorrow	0.26176
	3. Need You Now	0.27195
	4. Proud Of You	0.2727
	5. Underpressure	0.2872

Tabel 5-10. Tabel Hasil Uji Coba Skenario 3(2)

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Ghostbusters	1. Time Is Running Out	0.29078
	2. Makes Me Wonder	0.32015
	3. Need You Now	0.3286
	4. Close	0.33469
	5. There's No Tomorrow	0.33872
Pusing Pala Berbie	1. Proud Of You	0.21691
	2. You Are Still The One	0.21849
	3. All About That Bass	0.22594
	4. Because Of You	0.22745
	5. Goodbye My Lover	0.2438
Kopi Dangdut	1. Moliendo Cafe	0.14507
	2. Lust For Life	0.14923
	3. Makes Me Wonder	0.1678
	4. Time Is Running Out	0.179
	5. Creep	0.19103

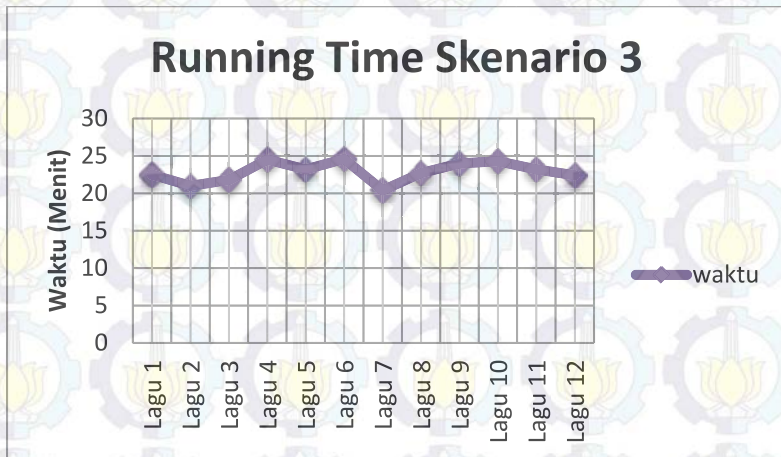
Tabel 5-11. Tabel Hasil Uji Coba Skenario 3(3)

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Bersamamu	1. Time Is Running Out	0.36233
	2. Close	0.36536
	3. Makes Me Wonder	0.37982
	4. Need You Now	0.39326
	5. Theres No Tomorrow	0.41768
Surat Undangan	1. I'll Be There	0.17847
	2. I Want A New Drug	0.2145
	3. All About That Bass	0.22335
	4. Because Of You	0.25395
	5. You Are Still The One	0.26799
Pejuang Cinta	1. Time Is Running Out	0.33846
	2. Makes Me Wonder	0.34792
	3. Close	0.36246
	4. There's No Tomorrow	0.36603
	5. Need You Now	0.37296

Tabel 5-12. Tabel Hasil Uji Coba Skenario 3(4)

Judul Lagu Inputan	Judul Keluaran Aplikasi	Nilai Jarak Kemiripan
Mahluk Tuhan Paling Sexy	1. Time Is Running Out	0.19589
	2. Makes Me Wonder	0.19975
	3. Moliendo Cafe	0.20555
	4. Lust For Life	0.21784
	5. Creep	0.25024
Surfin USA	1. Moliendo Café	0.15641
	2. Lust For Life	0.19312
	3. Makes Me Wonder	0.20362
	4. Time Is Runnin Out	0.25523
	5. Creep	0.26135
The Air That I Breath	1. Moliendo Cafe	0.1285
	2. Lust For Life	0.14823
	3. Creep	0.15169
	4. Makes Me Wonder	0.16907
	5. Sweet Little Sixteen	0.17849

Untuk menentukan kesesuaian keluaran dengan program dan analisa yang telah dibuat diperlukan pengujian akurasi dan *running time*. Perhitungan akurasi dilakukan menggunakan rumus 5-1. Dari hasil kelima uji coba maka didapatkan akurasi sebesar 83% karena 10 dari 12 percobaan yang berhasil dilakukan dengan benar. Hasil akurasi dari skenario ketiga cenderung lebih baik daripada skenario kedua. Hal ini disebabkan oleh plagiarisasi terhadap nada lagu asli yang dilakukan tanpa melakukan perubahan nada dasar, sehingga pendeteksian dapat dilakukan dengan baik. Sedangkan, pada skenario kedua aransemen terhadap lagu asli dilakukan dengan cara pengubahan nada dasar dari lagu asli dan perubahan penggunaan alat musik yang digunakan. Baiknya persentasi akurasi menunjukkan bahwa metode yang digunakan untuk melakukan pendeteksian kemiripan nada lagu dapat digunakan dan dapat dipercaya hasilnya. Grafik hasil *running time* skenario uji coba 1 dapat dilihat pada Grafik 5-3.



Grafik 5-3 Grafik Running Time Skenario 3

5.4. Analisis dan Evaluasi Hasil Uji Coba

Berdasarkan dari uji coba menunjukkan bahwa metode yang digunakan untuk membangun sistem dapat mendeteksi kemiripan nada lagu dengan baik. Hal ini ditunjukkan dengan nilai akurasi dan rata-rata *running time* yang didapatkan pada tiap-tiap skenario uji coba. Detail hasil akurasi dan rata-rata *running time* tiap skenario dijelaskan pada Tabel 5-13. Hasil akurasi ini didapatkan dengan menggunakan rumus 5-1. Detail dari penghitungan rata-rata *running time* tiap skenario dijelaskan pada Tabel 5-14 dan Tabel 5-15. Sedangkan, detail dari penghitungan akurasi dijelaskan pada dan Tabel 5-16 dan Tabel 5-17.

Tabel 5-13. Tabel Akurasi Uji Coba Dan *Running Time*

Skenario	Jumlah Uji Coba	Jumlah Uji Coba Berhasil	Jumlah Uji Coba Gagal	Akurasi	Rata-Rata <i>Running Time</i> (Menit)
1	3	3	0	100%	23.361
2	5	2	3	40%	24.03
3	12	10	2	83%	22.892
Rata-Rata <i>Running Time</i> Keseluruhan					23.428

Tabel 5-14. Tabel *Running Time* Tiap Skenario (1)

Skenario	Judul Lagu Masukan	<i>Running Time</i> (Detik)
1	Lust For Life	1274
	Do Somethin	1338
	Underpressure	1593
Rata-Rata <i>Running Time</i> (s)		1401.667

Tabel 5-15. Tabel *Running Time* Tiap Skenario (2)

Skenario	Judul Lagu Masukan	<i>Running Time</i> (Detik)
2	Proud Of You (Cover)	1471
	I'll Be There (Cover)	1290
	Good Bye My Lover (Cover)	1634
	Close (Cover)	1471
	You're Still The One (Cover)	1344
Rata-Rata <i>Running Time</i>		1442
3	Are You Gonna Be My Girl	1351
	It's Now Or Never	1260
	'O Sole Mio	1309
	Ghostbusters	1470
	Pusing Pala Barbie	1394
	Kopi Dangdut	1475
	Bersamamu	1226
	Surat Undangan	1360
	Pejuang Cinta	1442
	Mahluk Tuhan Paling Sexy	1459
	Surfin USA	1391
	The Air That I Breath	1345
Rata-Rata <i>Running Time</i>		1373.5

Tabel 5-16. Tabel Akurasi Uji Coba Skenario (1)

Skenario	Judul Lagu Masukan	Benar/ Salah (1/0)
1	Lust For Life	1
	Do Somethin	1
	Underpressure	1
Nilai Akurasi (berdasarkan rumus 5-1)		100%
2	Proud Of You (Cover)	1
	I'll Be There (Cover)	0
	Good Bye My Lover (Cover)	0
	Close (Cover)	1
	You're Still The One (Cover)	0
Nilai Akurasi (berdasarkan rumus 5-1)		40%

Akurasi 100% pada skenario pertama yang bertujuan untuk mengetahui fungsionalitas algoritma *dynamic time warping* menunjukkan bahwa penggunaan/pengimplementasian algoritma *dynamic time warping* sudah benar dan dapat dipakai untuk menghitung kemiripan nada lagu.

Akurasi yang kurang baik ditunjukkan pada skenario kedua, dimana pada skenario kedua, aplikasi diuji untuk menghitung kemiripan nada lagu dan mengeluarkan judul lagu yang dinyanyikan bukan oleh penyanyi aslinya. Akurasi 40% merupakan akurasi yang tidak baik. Hal ini dikarenakan pada lagu yang dinyanyikan bukan oleh penyanyi aslinya tersebut dinyanyikan pada nada dasar yang berbeda, disertai dengan

banyaknya aransemen pada nada-nada dasar dari lagu aslinya, dan juga penggunaan alat musik yang berbeda dari lagu aslinya. Perbedaan nada dasar penyanyi sangat mempengaruhi pendeteksian kemiripan nada lagu. Karena pada aplikasi yang dibangun ini belum dapat menghapus vokal penyanyi secara keseluruhan, mengingat lagu yang diinputkan bertipe *stereo* dan vokal suara pada tipe lagu *stereo* tidak dapat dihapus secara keseluruhan, tetapi hanya dapat disamarkan saja. Sehingga, nada dasar sang penyanyi dapat mengganggu akurasi dari aplikasi.

Tabel 5-17. Tabel Akurasi Uji Coba Skenario (2)

Skenario	Judul Lagu Masukan	Benar/ Salah (1/0)
3	Are You Gonna Be My Girl	1
	It's Now Or Never	1
	'O Sole Mio	1
	Ghostbusters	0
	Pusing Pala Berbie	1
	Kopi Dangdut	1
	Bersamamu	1
	Surat Undangan	1
	Pejuang Cinta	1
	Mahluk Tuhan Paling Sexy	1
	Surfin USA	0
	The Air That I Breath	1
Nilai Akurasi (berdasarkan rumus 5-1)		83%

Akurasi yang baik ditunjukkan pada skenario ketiga, dimana pada skenario ketiga, aplikasi diuji untuk menghitung kemiripan nada lagu dan mengeluarkan judul lagu yang diopinikan melakukan plagiarisasi. Akurasi 83% tergolong sudah baik. Akurasi ini didukung oleh plagiarisasi nada lagu yang dilakukan, biasanya tanpa mengubah nada dasar dari lagu aslinya. Hal ini membantu aplikasi untuk memberikan hasil dengan akurasi yang tinggi. Dengan baiknya nilai akurasi yang ditunjukkan pada skenario ketiga menunjukkan bahwa tujuan dari pembuatan aplikasi ini dapat tercapai, yaitu membantu dalam pendeteksian kemiripan nada lagu untuk mencegah plagiarisasi terhadap nada lagu.

Dari ketiga uji coba skenario, menunjukkan bahwa program berjalan lambat dan memiliki waktu rata-rata komputasi 23.428 menit. Hal ini disebabkan oleh komputasi pada pengestraksian nada lagu pada algoritma YAAPT. Besarnya dimensi yang dikerjakan pada algoritma tersebut sangat mempengaruhi dalam kecepatan aplikasi. *Dynamic Programming* pada YAAPT dan *dynamic time warping* juga berpengaruh pada kecepatan komputasi aplikasi, hal ini dikarenakan kurang handalnya Matlab dalam menangani *dynamic programming* sehingga sangat berpengaruh pada kecepatan, selain itu besarnya penggunaan *memory* RAM untuk penggunaan Matlab dan komputasi mempengaruhi aktivitas komputasi dan aktivitas komputer.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap pembuatan model, dapat diambil kesimpulan sebagai berikut:

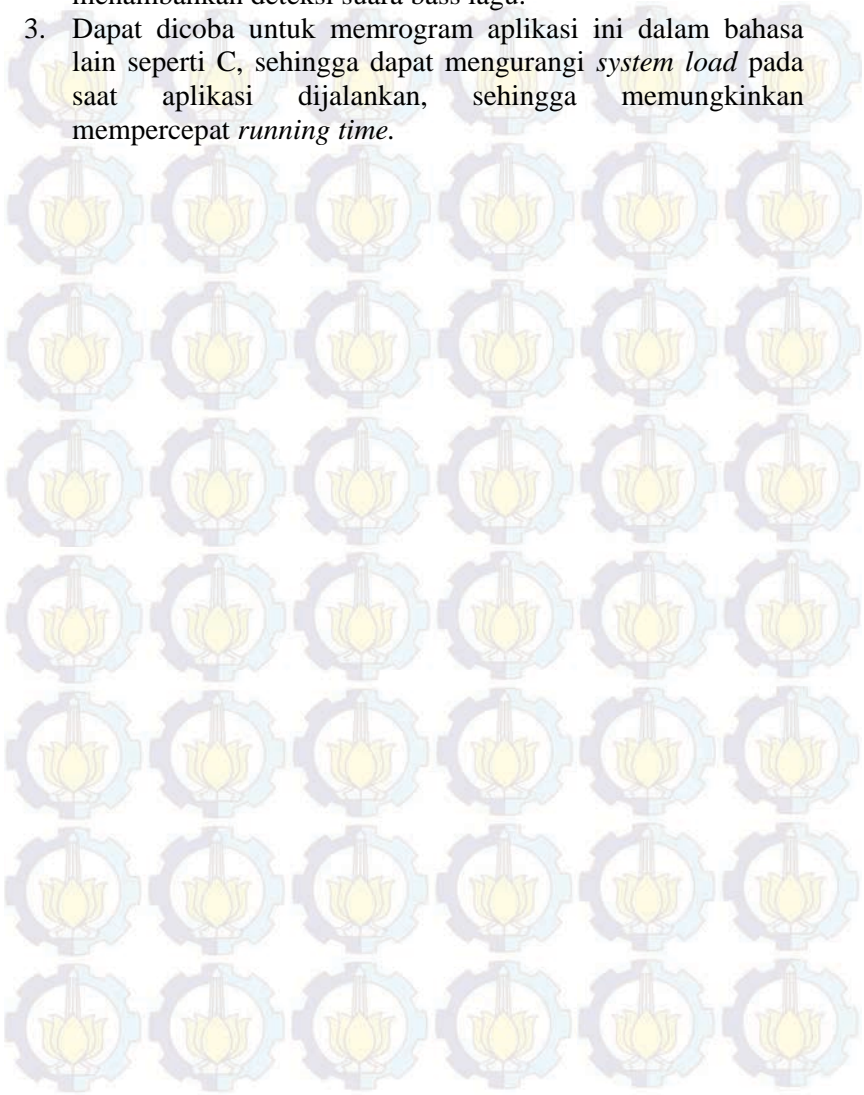
1. Metode YAAPT dapat melakukan ekstraksi fitur nada dari sebuah lagu dengan baik.
2. Metode *Dynamic Time Warping* dapat melakukan penghitungan kemiripan dengan baik.
3. Metode gabungan YAAPT dan *Dynamic Time Warping* dapat melakukan penghitungan kemiripan nada lagu dengan baik, dan mendapatkan hasil akurasi yang baik pula.
4. Lagu yang memiliki nada mirip atau sama, memiliki nilai jarak kemiripan antara 0 sampai 0,4 dengan penghitungan *Dynamic Time Warping* dan telah dinormalisasi.
5. Akurasi terbaik yang didapatkan adalah 83.33% dengan melakukan 12 uji coba terhadap lagu yang diopinikan melakukan plagiarisasi.
6. *Running time* rata-rata yang didapatkan dari seluruh skenario uji coba adalah 23.428 menit.

6.2. Saran

Saran yang diberikan untuk pengembangan aplikasi ini adalah:

1. Untuk meningkatkan nilai akurasi, dapat dicoba untuk menambahkan deteksi tempo lagu.

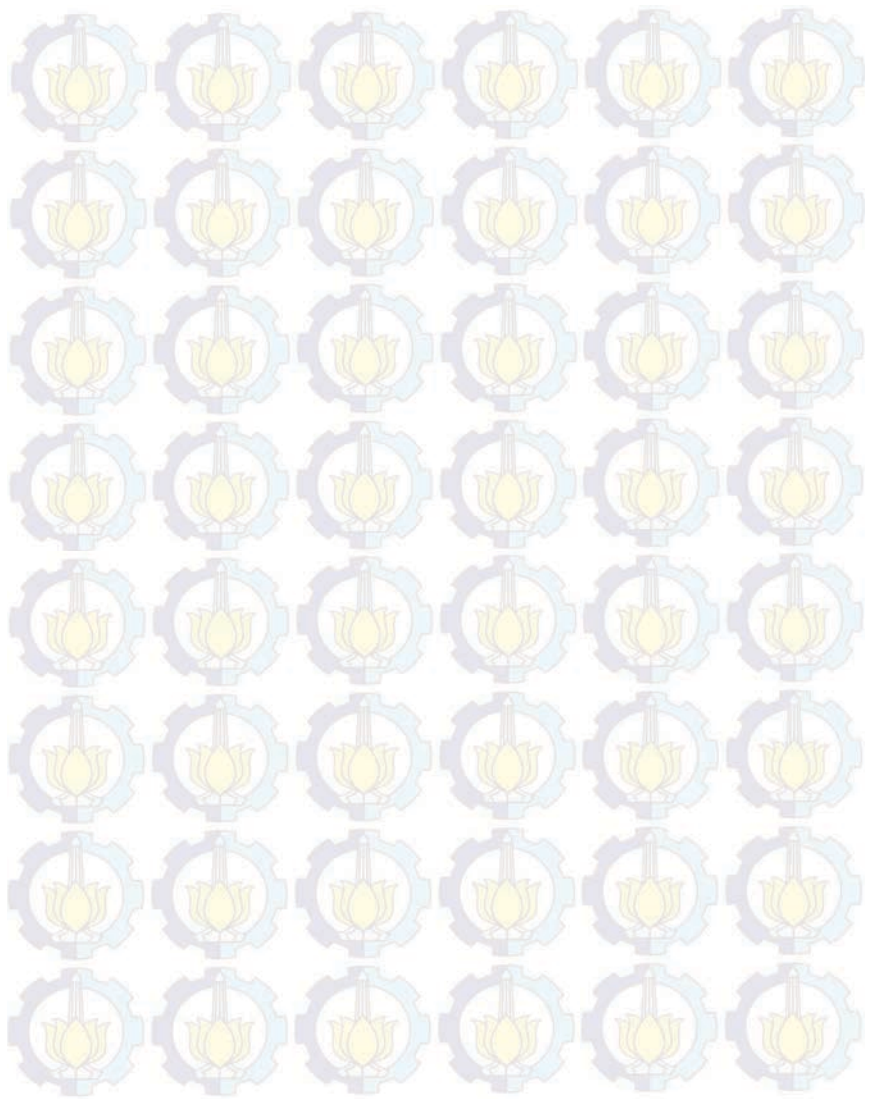
2. Untuk meningkatkan nilai akurasi, dapat dicoba juga untuk menambahkan deteksi suara bass lagu.
3. Dapat dicoba untuk memrogram aplikasi ini dalam bahasa lain seperti C, sehingga dapat mengurangi *system load* pada saat aplikasi dijalankan, sehingga memungkinkan mempercepat *running time*.



DAFTAR PUSTAKA

- [1] R. Arifanto, "rivayiarifanto.blogspot.com," 31 03 2014. [Online]. Available: <http://rivayiarifanto.blogspot.com/2014/03/pengertian-perbedaan-white-box-dan.html>. [Accessed 04 12 2014].
- [2] M. Said, "http://www.slideshare.net/Fajar_Sany/konsep-dasar-pengolahan-sinyal-digital," [Online]. Available: http://www.slideshare.net/Fajar_Sany/konsep-dasar-pengolahan-sinyal-digital. [Accessed 12 12 2014].
- [3] H. Oktavianto, "hary.okta's Blog," [Online]. Available: <https://haryoktav.wordpress.com/?=sinyal&searchbutton=go%21>. [Accessed 12 12 2014].
- [4] B. Szablsesi, "A History Of Melody," Barrie and Rockliff, London, 1965.
- [5] R. Y. Sipasulta, "Simulasi Sistem Pengacak Sinyal Dengan Metode FFT," Jurusan Teknik Elektro-FT UNSRAT, Manado, 2014.
- [6] K. Kasi and S. A. Zahorian, "Yet Another Algorithm For Pitch Tracking," Department of Electrical and Computer Engineering Old Dominion University, Norfolk, 2012.
- [7] D. Talkin, "A Robust Algorithm For Pitch Tracking," *Speech Coding and Synthesis*, pp. 495-518, 1995.
- [8] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimizaation for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 43-49, 1978.

[Halaman ini sengaja dikosongkan]



BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap pembuatan model, dapat diambil kesimpulan sebagai berikut:

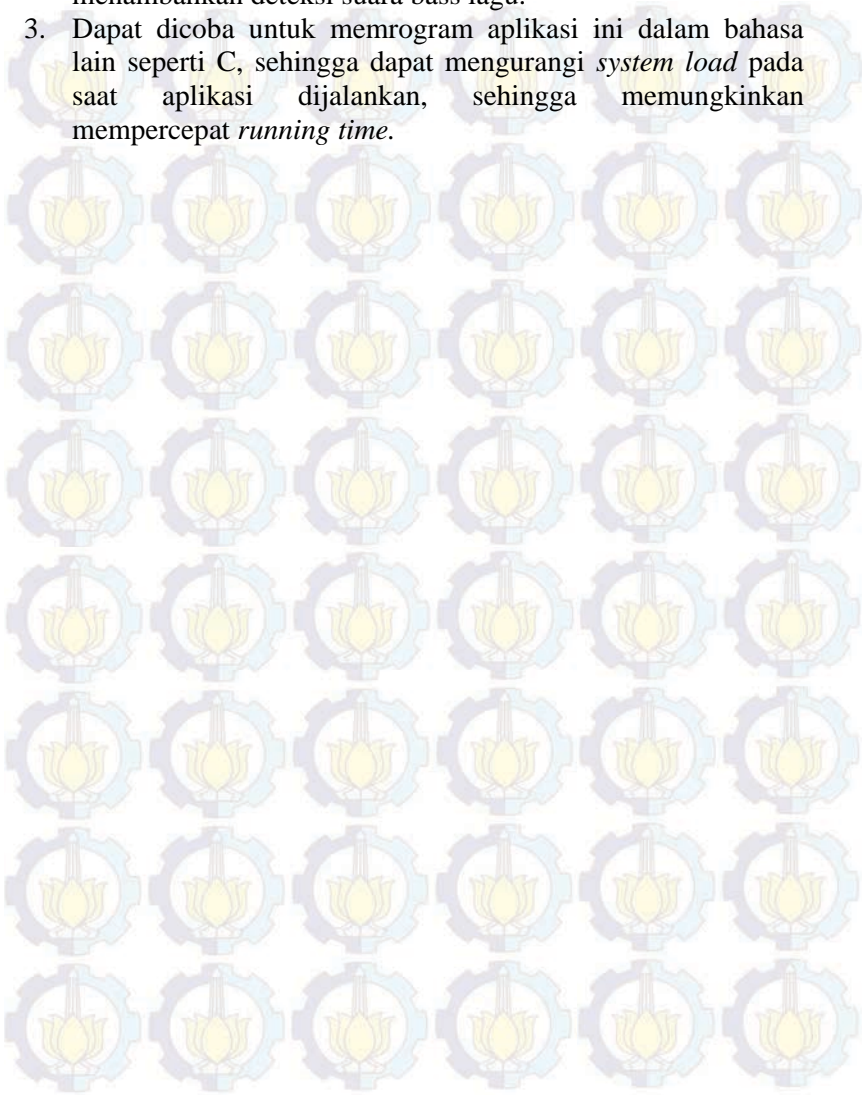
1. Metode YAAPT dapat melakukan ekstraksi fitur nada dari sebuah lagu dengan baik.
2. Metode *Dynamic Time Warping* dapat melakukan penghitungan kemiripan dengan baik.
3. Metode gabungan YAAPT dan *Dynamic Time Warping* dapat melakukan penghitungan kemiripan nada lagu dengan baik, dan mendapatkan hasil akurasi yang baik pula.
4. Lagu yang memiliki nada mirip atau sama, memiliki nilai jarak kemiripan antara 0 sampai 0,4 dengan penghitungan *Dynamic Time Warping* dan telah dinormalisasi.
5. Akurasi terbaik yang didapatkan adalah 83.33% dengan melakukan 12 uji coba terhadap lagu yang diopinikan melakukan plagiarisasi.
6. *Running time* rata-rata yang didapatkan dari seluruh skenario uji coba adalah 23.428 menit.

6.2. Saran

Saran yang diberikan untuk pengembangan aplikasi ini adalah:

1. Untuk meningkatkan nilai akurasi, dapat dicoba untuk menambahkan deteksi tempo lagu.

2. Untuk meningkatkan nilai akurasi, dapat dicoba juga untuk menambahkan deteksi suara bass lagu.
3. Dapat dicoba untuk memrogram aplikasi ini dalam bahasa lain seperti C, sehingga dapat mengurangi *system load* pada saat aplikasi dijalankan, sehingga memungkinkan mempercepat *running time*.



LAMPIRAN



The screenshot shows a window titled 'gui_TA' with the subtitle 'Aplikasi Deteksi Kemiripan Nada Lagu'. The test file 'IggyPop_Lust_For_Life.wav' is displayed in red. Below, a table lists five reference songs with their similarity scores.

No	Nama Lagu	Nilai Kemiripan
1.	IggyPop_Lust_For_Life	0
2.	MOLENDO_CAFE_Julio_Iglesias	0.1351
3.	Maroon5_Makes_Me_Wonder	0.1356
4.	Radiohead_Creep	0.19723
5.	Muse_Time_Is_Running_Out	0.2124


Gambar A1-1. Hasil Uji Coba Skenario 1 Lagu 1



The screenshot shows a window titled 'gui_TA' with the subtitle 'Aplikasi Deteksi Kemiripan Nada Lagu'. The test file 'BritneySpears_Do_Somethin.wav' is displayed in red. Below, a table lists five reference songs with their similarity scores.

No	Nama Lagu	Nilai Kemiripan
1.	BritneySpears_Do_Somethin	0
2.	TheChiffons_Hes_So_Fine	0.082923
3.	ChuckBerry_SweetLittleSixteen	0.11946
4.	MOLENDO_CAFE_Julio_Iglesias	0.1667
5.	Radiohead_Creep	0.1925

Gambar A1-2. Hasil Uji Coba Skenario 1 Lagu 2



The screenshot shows a window titled 'gui_TA' with a standard Windows interface. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test file 'QUEEN_UnderPressure.wav' is displayed in red. A table lists five songs with their similarity scores. The background features a repeating pattern of a yellow lotus flower inside a blue gear.

1.	QUEEN_UnderPressure	0
2.	Tony_Marlin_Theres_No_Tomorrow	0.21411
3.	NeedYouNow	0.2344
4.	Westlife_Close	0.26505
5.	YouAreStillTheOne	0.26565

Gambar A1-3. Hasil Uji Coba Skenario 1 Lagu 3



The screenshot shows a window titled 'gui_TA' with a standard Windows interface. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test file 'ProudOfYouCover.wav' is displayed in red. A table lists five songs with their similarity scores. The background features a repeating pattern of a yellow lotus flower inside a blue gear.

1.	ProudOfYou	0.24113
2.	YouAreStillTheOne	0.26611
3.	MeghanTrainor_All_About_That_Bass	0.26757
4.	GoodbyeMyLover	0.26673
5.	Jackson5_I_II_Be_There1	0.28426

Gambar A2-1. Hasil Uji Coba Skenario 2 Lagu 1



The screenshot shows a window titled 'gui_TA' with a yellow background and a repeating pattern of blue gears and yellow lotus flowers. The window contains the text 'Aplikasi Deteksi Kemiripan Nada Lagu' and 'Close - Westlife Cover.wav' in red. Below this is a list of five songs with their corresponding similarity scores.

No	Nama Lagu	Nilai Kemiripan
1.	Westlife_Close	0.34085
2.	Muse_Time_Is_Running_Out	0.34569
3.	Maroon5_Makes_Me_Wonder	0.35234
4.	Tony_Martin_Theres_No_Tomorrow	0.39086
5.	IggyPop_Lust_For_Life	0.39854

Gambar A2-2. Hasil Uji Coba Skenario 2 Lagu 2



The screenshot shows a window titled 'gui_TA' with a yellow background and a repeating pattern of blue gears and yellow lotus flowers. The window contains the text 'Aplikasi Deteksi Kemiripan Nada Lagu' and 'Elseo_Goodbye_my_Lover.wav' in red. Below this is a list of five songs with their corresponding similarity scores.

No	Nama Lagu	Nilai Kemiripan
1.	PictureBook	0.00046649
2.	Tony_Martin_Theres_No_Tomorrow	0.16842
3.	YouAreStillTheOne	0.18163
4.	NeedYouNow	0.2561
5.	ProudOfYou	0.28974

Gambar A2-3. Hasil Uji Coba Skenario 2 Lagu 3



Aplikasi Deteksi Kemiripan Nada Lagu		
Westlife_III_Be_There.wav		
1.	NeedYouNow	0.25087
2.	Tony_Martin_Theres_No_Tomorrow	0.25828
3.	Westlife_Close	0.26976
4.	Muse_Time_Is_Running_Out	0.3026
5.	QUEEN_UnderPressure	0.31103

Gambar A2-4. Hasil Uji Coba Skenario 2 Lagu 4



Aplikasi Deteksi Kemiripan Nada Lagu		
Lee Dewyze - You're Still The One Studio		
1.	Muse_Time_Is_Running_Out	0.35133
2.	Maroon5_Makes_Me_Wonder	0.35664
3.	IggyPop_Lust_For_Life	0.39828
4.	MOLENDO_CAFE_Juho_Iglesias	0.40205
5.	NeedYouNow	0.44956

Gambar A2-5. Hasil Uji Coba Skenario 2 Lagu 5



The screenshot shows a window titled 'gui_TA' with a standard Windows interface. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test file 'Jet_AreYouGonnaBeMyGirl.wav' is displayed in red. A list of five songs is shown, each with a rank number, the song name, and a similarity score.

Rank	Song Name	Similarity Score
1	IggyPop_Lust_For_Life	0.15331
2	MOLENDO_CAFE_Julio_Iglesias	0.15551
3	Maroon5_Makes_Me_Wonder	0.17403
4	Muse_Time_Is_Running_Out	0.19352
5	Radiohead_Creep	0.20856

Gambar A3-1. Hasil Uji Coba Skenario 3 Lagu 1



The screenshot shows the same 'gui_TA' application window, but with a different test file: 'elvispresley_its_now_or_never.wav'. The list of five songs and their similarity scores has changed.

Rank	Song Name	Similarity Score
1	Tony_Martin_Theres_No_Tomorrow	0.19269
2	YouAreStillTheOne	0.20756
3	NeedYouNow	0.29328
4	ProudOfYou	0.33129
5	QUEEN_UnderPressure	0.35986

Gambar A3-2. Hasil Uji Coba Skenario 3 Lagu 2



The screenshot shows a window titled 'gui_TA' with a light blue header. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu' (Song Similarity Detection Application). Below it, the reference song is 'Luciano Pavarotti - 'O sole mio.wav' in red text. A table lists five candidate songs with their similarity scores. The background features a repeating pattern of blue gears and yellow lotus flowers.

No	Nama Lagu	Nilai Kemiripan
1.	YouAreStillTheOne	0.24588
2.	Tony_Martin_Theres_No_Tomorrow	0.26176
3.	NeedYouNow	0.27195
4.	ProudOfYou	0.2727
5.	QUEEN_UnderPressure	0.2872

Gambar A3-3. Hasil Uji Coba Skenario 3 Lagu 3



The screenshot shows the same 'gui_TA' application window, but with a different reference song: 'Ghostbusters.wav' in red text. The table lists five candidate songs with their similarity scores. The background pattern of blue gears and yellow lotus flowers remains the same.

No	Nama Lagu	Nilai Kemiripan
1.	Muse_Time_Is_Running_Out	0.29076
2.	Maroon5_Makes_Me_Wonder	0.32015
3.	NeedYouNow	0.3286
4.	Westlife_Close	0.33469
5.	Tony_Martin_Theres_No_Tomorrow	0.33872

Gambar A3-4. Hasil Uji Coba Skenario 3 Lagu 4



The screenshot shows a window titled 'gui_TA' with a background of blue gears and yellow lotus flowers. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test subject is 'Puteri Bahar, Pusing Pala Barbie (Jelita)'. A list of five songs is displayed with their corresponding similarity scores.

No	Nama Lagu	Nilai Kemiripan
1.	ProudOfYou	0.21691
2.	YouAreStillTheOne	0.21849
3.	MeghanTrainor_Ali_About_That_Bass	0.22594
4.	BecauseOfYou	0.22745
5.	GoodbyeMyLover	0.2438

Gambar A3-5. Hasil Uji Coba Skenario 3 Lagu 5



The screenshot shows a window titled 'gui_TA' with a background of blue gears and yellow lotus flowers. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test subject is 'The Sallys - Kopi Dangdut.wav'. A list of five songs is displayed with their corresponding similarity scores.

No	Nama Lagu	Nilai Kemiripan
1.	MOLENDO_CAFE_Julio_Iglesias	0.14507
2.	IggyPop_Lust_For_Life	0.14923
3.	Maroon5_Makes_Me_Wonder	0.1678
4.	Muse_Time_Is_Running_Out	0.179
5.	Radichead_Creep	0.19103

Gambar A3-6. Hasil Uji Coba Skenario 3 Lagu 6



Aplikasi Deteksi Kemiripan Nada Lagu		
Vierra_Bersamamu.wav		
1.	Muse_Time_Is_Running_Out	0.36233
2.	Westlife_Close	0.36538
3.	Maroon5_Makes_Me_Wonder	0.37982
4.	NeedYouNow	0.39326
5.	Tony_Martin_Theres_No_Tomorrow	0.41768

Gambar A3-7. Hasil Uji Coba Skenario 3 Lagu 7



Aplikasi Deteksi Kemiripan Nada Lagu		
Diah Iskandar - Surat Undangan { by Sonny		
1.	Jackson5_I_If_There1	0.17847
2.	Huey_Lewis_and_the_News_I_want_a_new_drug	0.2145
3.	MeghanTrainor_All_About_That_Bass	0.22335
4.	BecauseOfYou	0.25395
5.	YouAreStillTheOne	0.26799

Gambar A3-8. Hasil Uji Coba Skenario 3 Lagu 8



The screenshot shows a window titled 'gui_TA' with a standard Windows interface. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test file is identified as 'Hello - Pejuang Cinta (HD) wav' in red text. A list of five songs is displayed, each with a rank number, the song name, and a similarity score.

Rank	Song Name	Similarity Score
1	Muse_Time_Is_Running_Out	0.33846
2	Maroon5_Makes_Me_Wonder	0.34792
3	Weslife_Close	0.36246
4	Tony_Martin_Theres_No_Tomorrow	0.36603
5	NeedYouNow	0.37296

Gambar A3-9. Hasil Uji Coba Skenario 3 Lagu 9



The screenshot shows a window titled 'gui_TA' with a standard Windows interface. The main title is 'Aplikasi Deteksi Kemiripan Nada Lagu'. Below it, the test file is identified as 'Mulan Jameela - Makhluq Tuhan Paling' in red text. A list of five songs is displayed, each with a rank number, the song name, and a similarity score.

Rank	Song Name	Similarity Score
1	Muse_Time_Is_Running_Out	0.19589
2	Maroon5_Makes_Me_Wonder	0.19975
3	MOLEND0_CAFE_Julio_Iglesias	0.20555
4	IggyPop_Lust_For_Life	0.21784
5	Radiohead_Creep	0.25024

Gambar A3-10. Hasil Uji Coba Skenario 3 Lagu 10



Gambar A3-11. Hasil Uji Coba Skenario 3 Lagu 11



Gambar A3-12. Hasil Uji Coba Skenario 3 Lagu 12

Tabel A1-1. Tabel Lagu Yang Diopinikan Melakukan Plagiarisasi

Tabel Lagu Yang Diopinikan Plagiat			
Judul Lagu Asal	Judul Lagu Yang Diopinikan Plagiat	Sumber	Nilai Jarak Kemiripan
Lust For Life	Are You Gonna Be My Girl	http://listverse.com/2010/08/23/top-10-popular-music-rip-offs/	0.15331
There's No Tomorrow	Its Now Or Never	http://forums.stevetehoffman.tv/threads/famous-borrowed-or-plagiarized-melodies.274901/	0.19269
There's No Tomorrow	O Sole Mio	http://forums.stevetehoffman.tv/threads/famous-borrowed-or-plagiarized-melodies.274901/	0.26176
Creep	The Air That I Breathe	Wardle, Ben. "Get off Coldplay's case – similar songs can co-exist peacefully". Guardian.co.uk. 12 May 2009. Retrieved on 21 April 2013.	0.15169
All About That Bass	Pusing Pala Barbie	http://entertainment.kompas.com/read/2015/02/11/123118610/Pusing_Pala_Barbie.99.Persen.Jiplak.All.About.That.Bass.	0.22594
Moliendo Cafe	Kopi Dangdut	https://en.wikipedia.org/wiki/Moliendo_Caf%C3%A9	0.14507
Close	Bersamamu	http://www.lautanindonesia.com/forum/index.php?topic=18871.235;wap2	0.36536
I'll Be There	Surat Undangan	http://forum.gemscool.com/post/list_forum?cigrsq=34&threadSq=93409	0.17847
Makes Me Wonder	Pejuang Cinta	http://forum.lyogame.com/archive/index.php/t-44712.html?s=-4577106c6154b59095d5432ee9819169	0.34792
Time Is Running Out	Mahluk Tuhan Paling Sexy	http://forum.lyogame.com/archive/index.php/t-44712.html?s=-4577106c6154b59095d5432ee9819169	0.19589
Sweet Little Sixteen	Surfin USA	Pegg, Bruce. <i>Brown Eyed Handsome Man</i> (2002): 162–163	0.31751
I Want A New Drug	Ghostbusters	http://www.billboard.com/articles/news/list/6501950/songs-accused-plagiarism-no-1-hot-100-blurred-lines	0.76954

[Halaman ini sengaja dikosongkan]

